Docusnap 6.1 - Configuration Manual

English Version

Table of Contents

Part

1 Introduction

We have pre-configured Docusnap 6.1 for the majority of tasks required by our users. If you need features that add to the default configuration, Docusnap provides a number of options for this purpose:

- Changing and creating reports
- Adding additional columns and tables to the Docusnap database
- Defining additional views for data evaluation
- Defining additional structures in the Data Explorer, License Management, Permission Analysis and Organization tree views
- Creating new or extending existing data entry screens

To perform any of the tasks mentioned above, except for the first one (i.e. changing and creating reports), you need a license for the Docusnap Customizing module. Modified settings can easily be transferred to other Docusnap installations.

Important note: With the exception of reports, the customizations in Docusnap are limited to adding new elements to existing structures or defining new elements. Elements that already exist in Docusnap cannot be deleted, since this would lead to uncontrollable application behavior.

This manual is subdivided into six sections:

Part 1: File Structure of a Docusnap Installation

Docusnap uses certain directory structures and database information to represent custom settings. This section introduces these structures and mechanisms and serves as the basis for the following sections.

Part 2: Reporting

This section explains how to create new reports, use the Docusnap Report Designer and assign reports to an object in the tree view.

Part 3: Database Structures

This section illustrates the structure of the Docusnap database and shows you how to extend the existing data structures.

Part 4: Meta Objects

Meta objects represent individual elements in the four different tree hierarchies. For information on how to define or extend existing meta objects, see part 3 of this

manual.

Part 5: Data Entry Screens

Data entry screens are user interface elements that allow the users to enter data. You can extend existing data entry screens or define new ones.

Part 6: Distributing Customizations

In Docusnap, you can export your customizations and import them to other environments. This way, it is possible to make customizations of the database structure, the meta objects and data entry screens available to other databases or even other Docusnap installations without much effort.

1.1 Conventions

In order to improve the readability of this document, the following conventions apply:

Normal text uses the Calibri font. Names of buttons, checkboxes, etc. are written in *italics*.

Code samples are formatted in `Courier New`.

Some sections feature tips for using Docusnap. These tips are indicated by a light bulb.

Warning sections are highlighted by a warning sign. Warnings refer to issues that should be taken into consideration when working with Docusnap.

Text that contains additional information is highlighted by an information sign.

1.2 Organization of a Docusnap Installation

Docusnap 6.1 is a multi-user software that requires a certain network directory structure in order to function properly. As part of the default Docusnap installation process, the following directories will be created in the program directory:

\bin	Tools and script files

\DataEdit	Data entry screen definition files (.des) defined by the software manufacturer
\Reporting	Predefined report files (.mrt) in German and English, and styles including report assignments (.xml)
\Help	Help system
\Schema	Files defining database structures, the ADS scanning process, and other definition files (*.dss,*.xml,*.xsd)
\De	Language resource files for the German user interface
\En	Language resource files for the English user interface
\Templates	Visio stencil files (*.vss) for plan creation
\Mibs	Standard MIB files for the SNMP scanning process
\Dictionaries	Spell Check Dictionaries for IT Concept
\Design	Design Schema
\ITConceptTemplates	Predefined Templates for IT Concepts
\Tools	Executable Files for DocusnapScript and DocusnapLink

Please note the following: All these directories have been defined by the software manufacturer and the corresponding files are provided automatically during the Docusnap installation or update process. Under no circumstances change these directories, since otherwise, the proper execution of Docusnap cannot be guaranteed.

As part of the initial Docusnap configuration, each user must specify a local settings folder. Once you have specified this folder, Docusnap automatically copies the contents of the Reporting directory to the local settings folder and simultaneously creates the \DataEdit and \Schema directories.

If you also specify a team settings folder, the same directories will be created in that folder as well. Both copy processes will only be performed if the target folders are empty.

In the event that custom modifications have been made, the definition files will always be saved in the team settings folder, or, if it is not available, in the local settings folder. Docusnap will never create custom definition files within the program directory.

When Docusnap starts, it first checks if a team settings folder is available. If so, the report and definition files from that folder will be used. If a team settings folder has not been specified or the network path cannot be reached, the local settings folder will be used. If that folder is not available either, the predefined settings from the program directory will be used.

If you save user-defined settings to a file, the letter "u" will always be appended to the last letter of the file extension. For example, definition files for custom data entry screens always have a .deu file extension.

The reporting system represents an exception to this rule, since we constantly supply new reports. For that reason, the Docusnap Update process will store the new reports in the \Reporting directory under the local program directory. At the next startup, Docusnap will discover that new reports are available and then prompt you whether you want to copy them to the corresponding settings folder(s) . If existing reports have been modified, they will be overwritten in this case. If you intend to customize a report, it would be a good idea to create a copy, rename it and then make your changes to the renamed copy. This ensures that customized reports will not be overwritten.

Part

2 Reporting System

The Docusnap database stores information that has either been detected by the network inventory scan or manually entered by the user.

Docusnap provides reports for evaluating and printing this information. For example, you can create a report that contains information about a single system or an overview of a domain. The reports can be exported to documents in various formats (e.g. docx, pdf, html, odt), printed or sent by e-mail.

Reports can be executed from various levels in the tree view. Docusnap provides predefined reports which process existing data from the database.

You can select a global report format in the *Designs* dialog. These format settings will be used for all companies. You can customize the company logos, colors and fonts for the reports so that they reflect the corporate identity. Report formatting can be customized even further at the company level.

What is more, you can edit existing reports and create new ones.

2.1 Basics

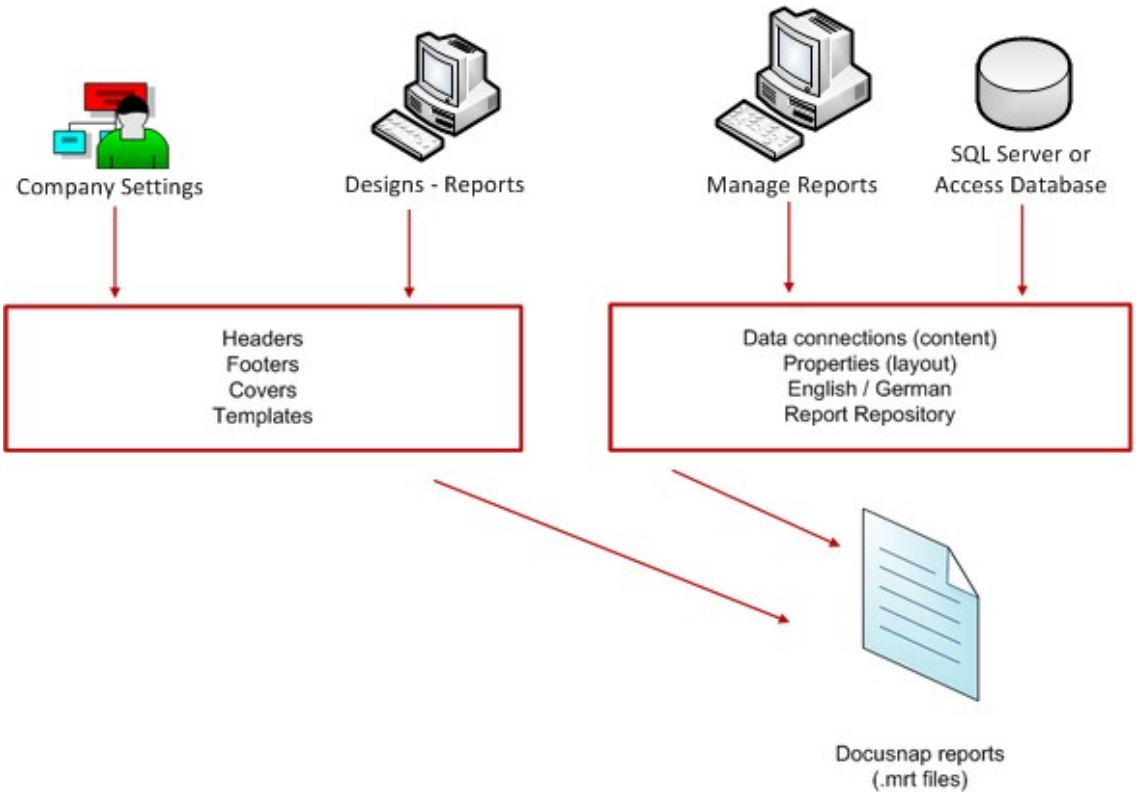
All information stored in the database can be output in reports, filtered and sorted as required.

For each report, additional information can be provided on a cover page, in a header and/or a footer. For each company, you can design a custom cover page and an individual header and footer. The styles used for the report can also be created centrally for all companies. However, you can also customize them at the company level.

The predefined reports are stored in the program directory when Docusnap is installed. After you have selected a local and/or team settings folder, all reports are copied to the *Reporting* folder. This folder is the report repository from which the reports will be retrieved later.

To generate/execute a report, click it in the tree view. The data is displayed on the Reports tab of the main window. Depending on the settings, the report consists of the header and footer, the cover page and the actual report content. The report format is controlled by styles. The content of the report will be retrieved from the database. In the *Manage Reports* dialog, you can determine the position of the report in the tree structure. In addition, this dialog allows you specify the language for the report and its properties, such as author or description.

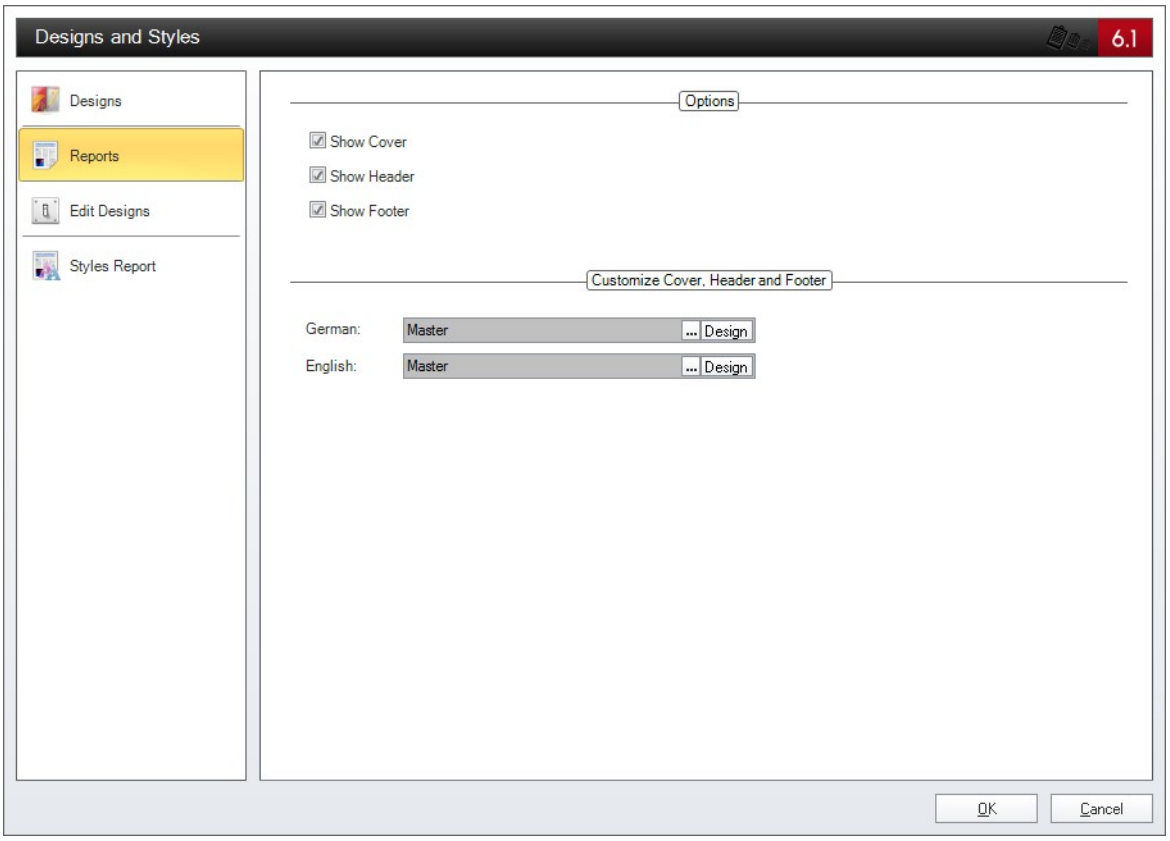
After the report has been executed, it can be printed or exported to any file format desired (docx, pdf, html, odt, etc).



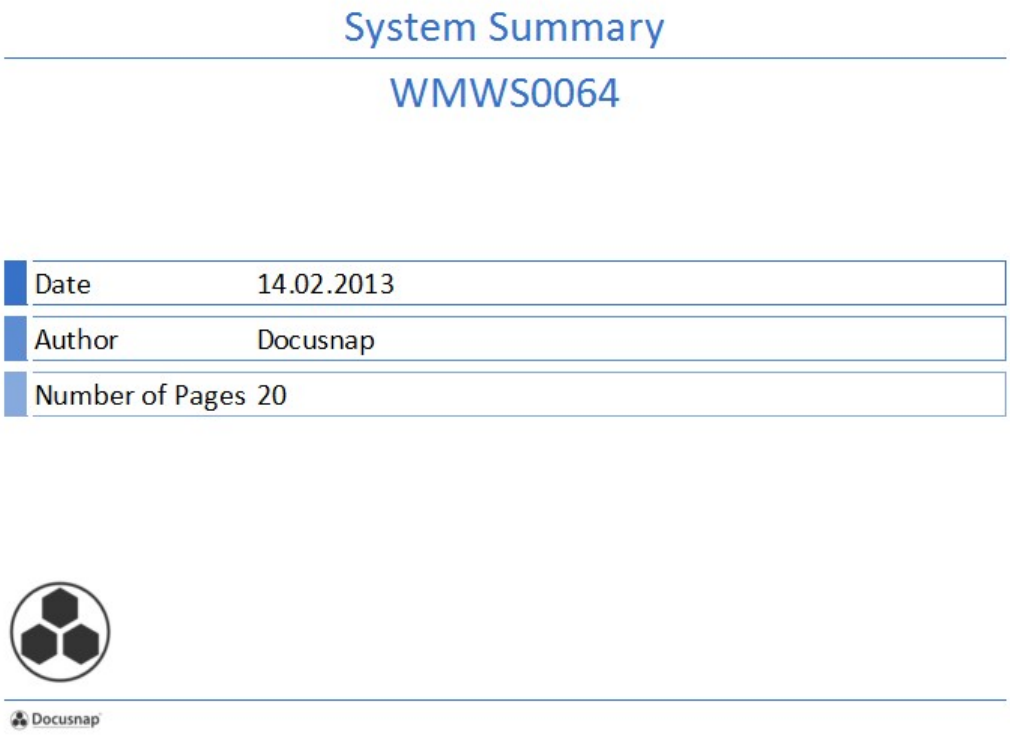
2.2 Report Structure

Additional information, such as the report name, the author or the creation date, is displayed on the cover page, as well as in the header and footer. This information can be changed from the *Manage Reports* dialog.

To select the desired cover pages, header and footer, go to the *Designs and Styles* or *Define Company Settings* dialog. In order to enable Docusnap to generate reports either in English or in German, the report which defines the cover page, header and footer is stored twice, once for English and once for German.



The cover page shows the report name and the tree view meta object for which the report was generated. What is more, the current date, the name of the author and the page count are also shown. The name of the author and the report name can be changed from the *Manage Reports* dialog. If you enter a description in that dialog, it will also appear on the cover page.


System Summary

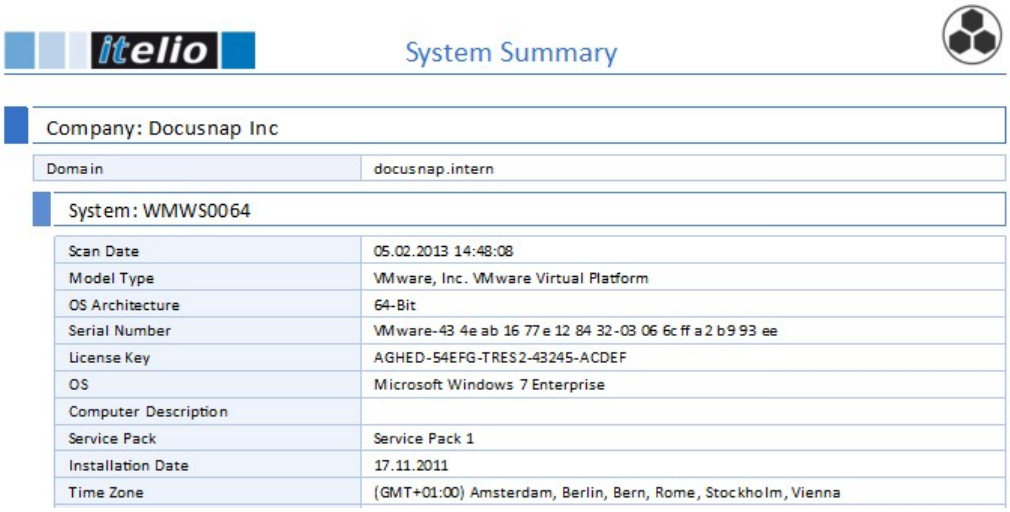
WMWS0064

Date	14.02.2013
Author	Docusnap
Number of Pages	20

Docusnap

The header shows the report name and two logos. Both logos can be selected from the *Designs and Styles* or the *Define Company Settings* dialog.

The page count and the underlying tree object for the report are shown in the footer.


itelio

System Summary

Company: Docusnap Inc

Domain	docusnap.intern

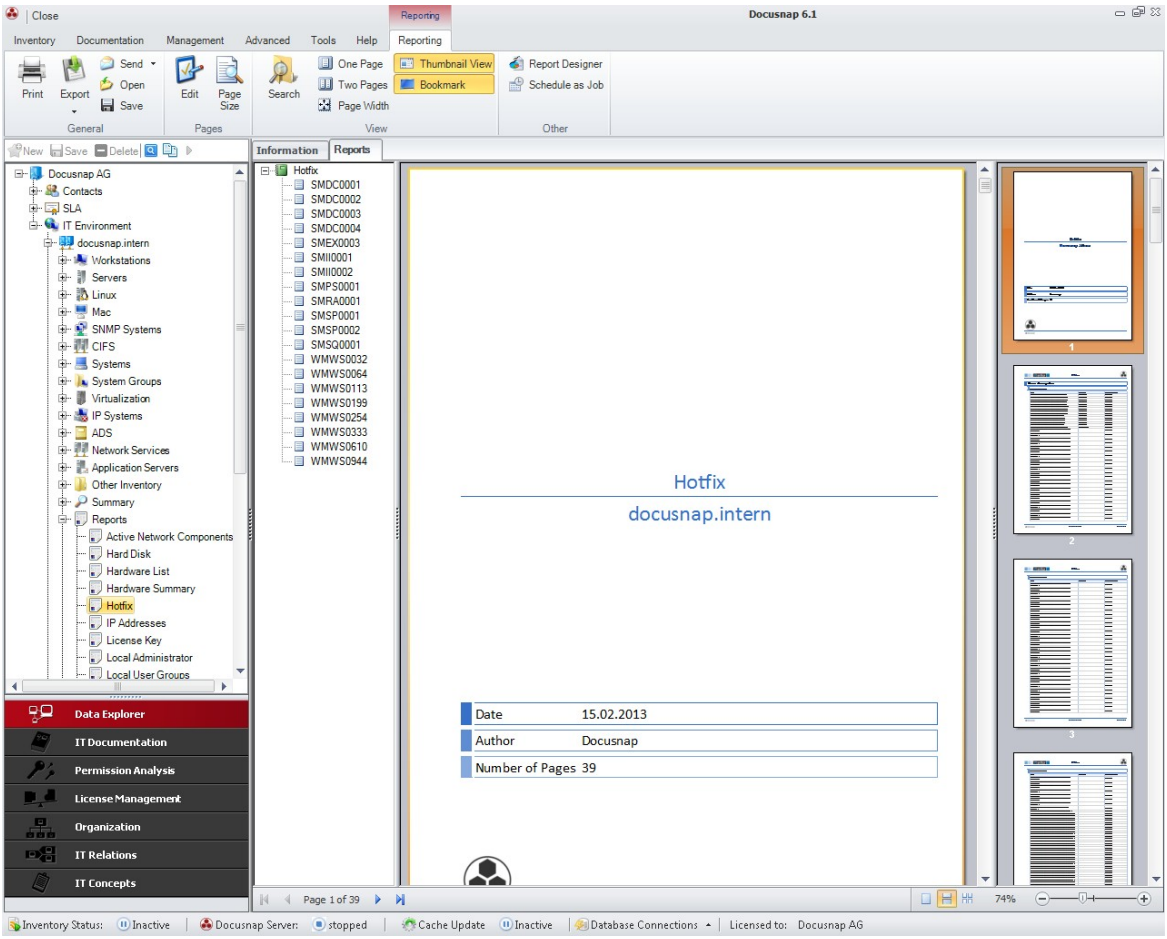
System: WMWS0064

Scan Date	05.02.2013 14:48:08
Model Type	VMware, Inc. VMware Virtual Platform
OS Architecture	64-Bit
Serial Number	VMware-43 4e ab 16 77 e 12 84 32-03 06 6c ff a2 b9 93 ee
License Key	AGHED-54EFG-TRES2-43245-ACDEF
OS	Microsoft Windows 7 Enterprise
Computer Description	
Service Pack	Service Pack 1
Installation Date	17.11.2011
Time Zone	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna

2.3 Output

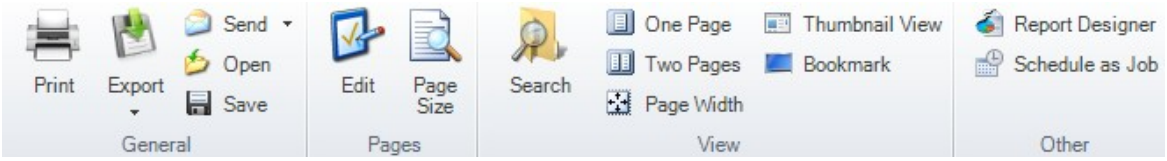
Reports are usually shown under the Reports node in the tree view. Click the desired report to display in on the *Reports* tab of the main window. A thumbnail

can be viewed to the right of the window. If bookmarks have been defined for the report, they will be displayed in the left pane of the window.



When you generate/execute a report, an additional ribbon displays. It contains buttons to modify the view and to export, print or edit the report.

When you click the *Designer* button, the current report will be opened in the Report Designer for editing.



2.4 Report Management

Reports can be managed from the *Manage Reports* dialog. This dialog allows you to create new reports and edit existing ones. Each report has properties, such as a name, author and language. These properties can also be edited from this dialog.

All reports are listed in the tree view. The tree view consists of meta objects. For a detailed description of meta objects, see the meta objects section. Each report can be linked with one or more meta objects. This determines the position of the report

in the tree structure. The link can be defined in the *Manage Reports* dialog.

Data can only be shown in the tree view if the corresponding meta objects have been linked with tables. Each table in the database has a primary key. The primary key of a table is a value that uniquely identifies each record in the table. When you generate/execute a report, the primary key of the associated table will be passed to the report.

Reports can be linked with any meta objects that exist in the database. If you link the report with a meta object of the Data type, the primary key of that table will be passed to the report. For meta objects of a different type, Docusnap will always pass the primary key of the next parent object which is linked with a table.

Thereby, this value can be used for filtering and only data associated with that object will be included in the report. For example, if the primary key for the domain is passed and used for filtering, the report will only show data related to that domain.

In the Manage Reports dialog, all reports that have been created are listed both in German and English. In the drop-down list, you can select the language in which the report will be displayed. In the left pane of the dialog, the available reports are listed. The properties of the selected report are displayed in the right pane.

The author of all reports created by Docusnap defaults to Docusnap. To change the author in some reports or delete a number of reports in one go, you can select multiple reports at the same time.

All of the reports can be deleted or renamed. However, care should be taken with some reports.
Several Docusnap features are based on reports. If you delete or rename one of these reports, those features might no longer work properly.

The datasheets and overviews you can create in the IT Documentation module are based on reports. The output from data comparisons is also provided by means of a report. The effective permissions calculation results are displayed in a report as well. Therefore, do not delete or rename the following reports:

Documentation Reports

Datenblatt Arbeitsstation	Workstation Datasheet
Datenblatt Server	Server Datasheet
Datenblatt Mac	Mac Datasheet
Datenblatt Linux	Linux Datasheet
Datenblatt SNMP	SNMP Datasheet
Datenblatt AD Standorte	AD Sites Datasheet
Datenblatt CIFS	CIFS Datasheet

Datenblatt Datenspeicher	Data Store Datasheet
Datenblatt Email Kontakte	Email Contacts Datasheet
Datenblatt Hyper-V	Hyper-V Datasheet
Datenblatt IIService	IIService Datasheet
Datenblatt Netzwerk	Network Datasheet
Datenblatt öffentliche Ordner DB	Public Folders DB Datasheet
Datenblatt öffentliche Ordner Gruppen	Public Folders Groups Datasheet
Datenblatt öffentlichen Ordner	Public Folders Datasheet
Datenblatt Organisation	Organization Datasheet
Datenblatt Postfächer	Mailboxes Datasheet
Datenblatt Postfächer Datenbank	Mailboxes DB Datasheet
Datenblatt Postfächer Gruppen	Mailboxes Groups Datasheet
Datenblatt Serverkonfiguration	Server Configuration Datasheet
Datenblatt SQL Datenbank	SQL Database Datasheet
Datenblatt SQL Server	SQL Server Datasheet
Datenblatt Verteilergruppen	Distribution Groups Datasheet
Datenblatt Virtuelle Maschine	Virtual Machine Datasheet
Datenblatt VMware Server	Virtual Machine Server Datasheet

Data Comparison Report

Vergleichsdaten	Compare Objects

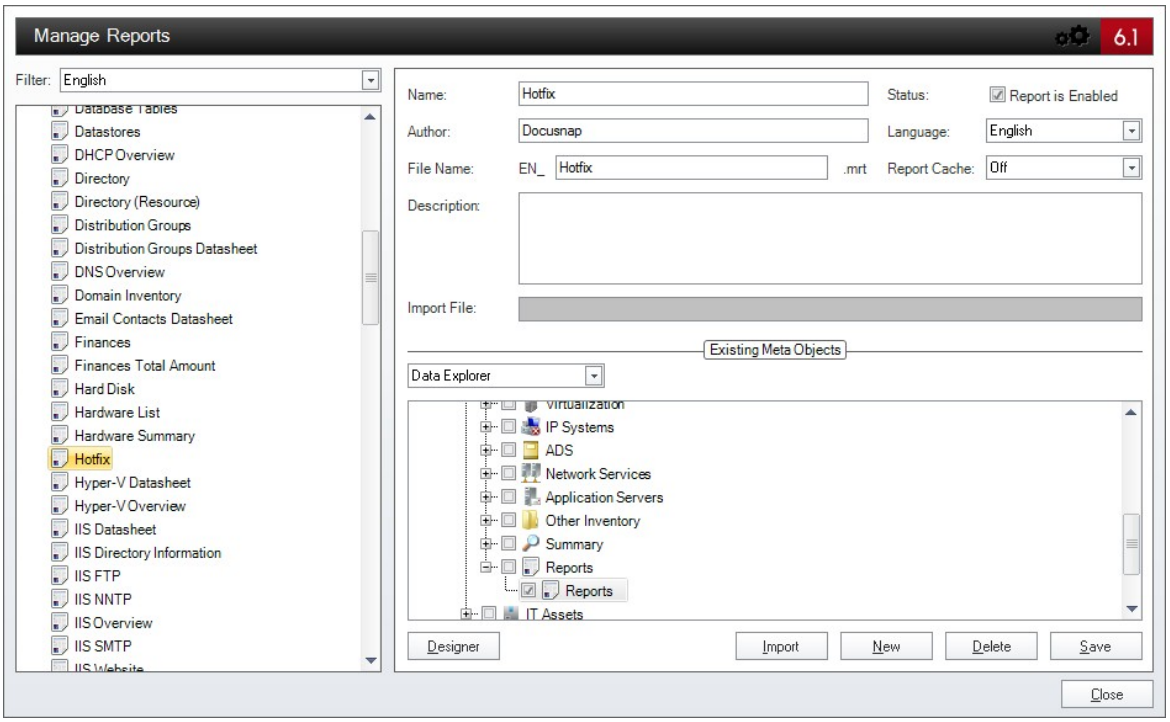
Effective Permissions Report

Verzeichnis (Ressource)	Directory (Resource)
Benutzer (Ressource)	User (Resource)
Berechtigungsanalyse Aktuelle Ansicht	Permission Analysis - Current View

Report Properties	
Name	The name specified in the Name field is displayed in the report list of this dialog and in the tree view.
Status	The current report will only be visible in the tree view if the *Report is Enabled* checkbox is ticked. If this checkbox is empty, the current report will not be displayed in the tree view.
Author	This field shows the name of the person who created the report. This name will be shown under Author on the

	report cover page.
Language	Reports for which the German language has been selected will only be displayed if the Docusnap language setting is also German. English reports are only displayed if the language has been set to English. The language that you select in the Manage Reports dialog determines the file name prefix, i.e. either "DE_" or "EN_". After creating and saving the report, you can still change the language. If you switch the language, e.g. from German to English, the prefix will be changed from "DE_" to "EN_" and the report will be displayed as soon as the display language for Docusnap is changed to English. The headings in the report, however, remain in the language in which they were created.
File Name	The file name for the reports is composed of the "DE_" or "EN_" prefix, the report name and the ".mrt" file extension. The prefix depends on the selected language. You can choose any name you like, but it is recommended to enter a meaningful name. This file name will be used to save the report on the disk.
Report Cache	If you want to create extensive reports, you need to enable the *Report Cache* feature. This feature caches the pages of the report. This makes sense in case the RAM on your machine would not be sufficient to create the report. If you select the *Auto* setting here, the report will be split once it has reached 500 pages. The *On* setting causes Docusnap to cache the pages of the report from the first page on. Upon completion of the creation process, the pages will be combined into a single report. This step takes additional time, so make sure to only select this setting if your RAM is insufficient for creating the report. It is recommended to use the *Auto* setting.
Description	The text entered here will appear on the cover page. This field is optional, i.e. you can save the report without a description.

Existing Meta Objects	This group in the lower half of the dialog displays the meta objects existing in the tree views. Select the desired tree view from the drop-down list. Enable the checkbox of the meta object you want to link the report with. The reports will be listed in the tree section below the node of the meta object they are linked with. If you link the report with a meta object of the Report type, the report will replace this meta object in the tree view. If desired, you can change the object the report is linked with and you can link the same report with multiple nodes in the tree



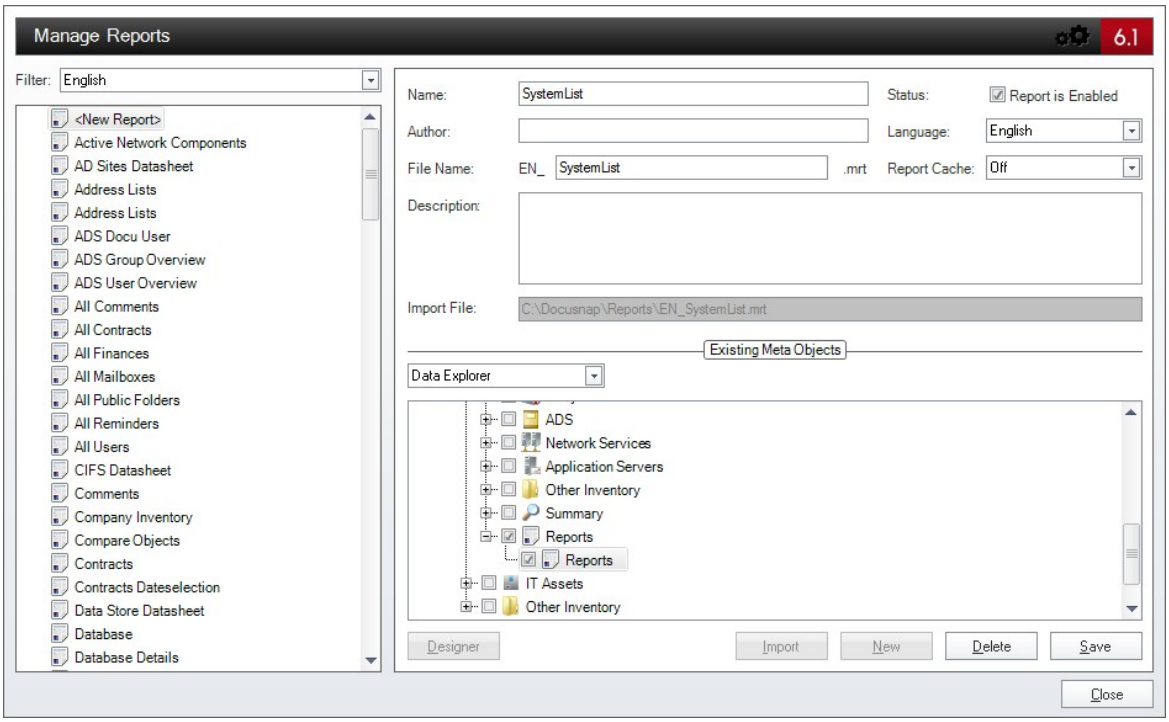
When you click the *Designer* button at the bottom of the dialog, the selected report will open in the Designer where it can be edited. By clicking the *New* button, you create a new, empty report. Enter all required properties and then click *Save* to apply your entries. To delete reports that are no longer needed both from this list and from the hard disk, click the *Delete* button.

2.5 Report Import

In addition to managing existing reports, the *Manage Reports* dialog can also be used to import reports. If you would like multiple users to access the same reports while working in Docusnap, use of the *Team Settings* folder is recommended. This feature is required if you need to import new reports, e.g. when the software manufacturer provides a new report that was created specifically for a company.

To select a report file (*.mrt), click the *Import* button. The file name will be used as the report name. Both the report name and the file name can be changed. The report can only be saved if you have specified its author. To display the report in the tree view, you need to select a meta object. When you click the *Save* button, the imported report will be saved to the *Reporting* folder under the *Team or Local Settings* folder.

If a report with the same file name already exists, you can either save the report under a different file name or overwrite the existing report with the imported report.



2.6 Report Designer

The Report Designer allows you to open predefined reports for editing or create new reports.

There are two ways to open the Report Designer:

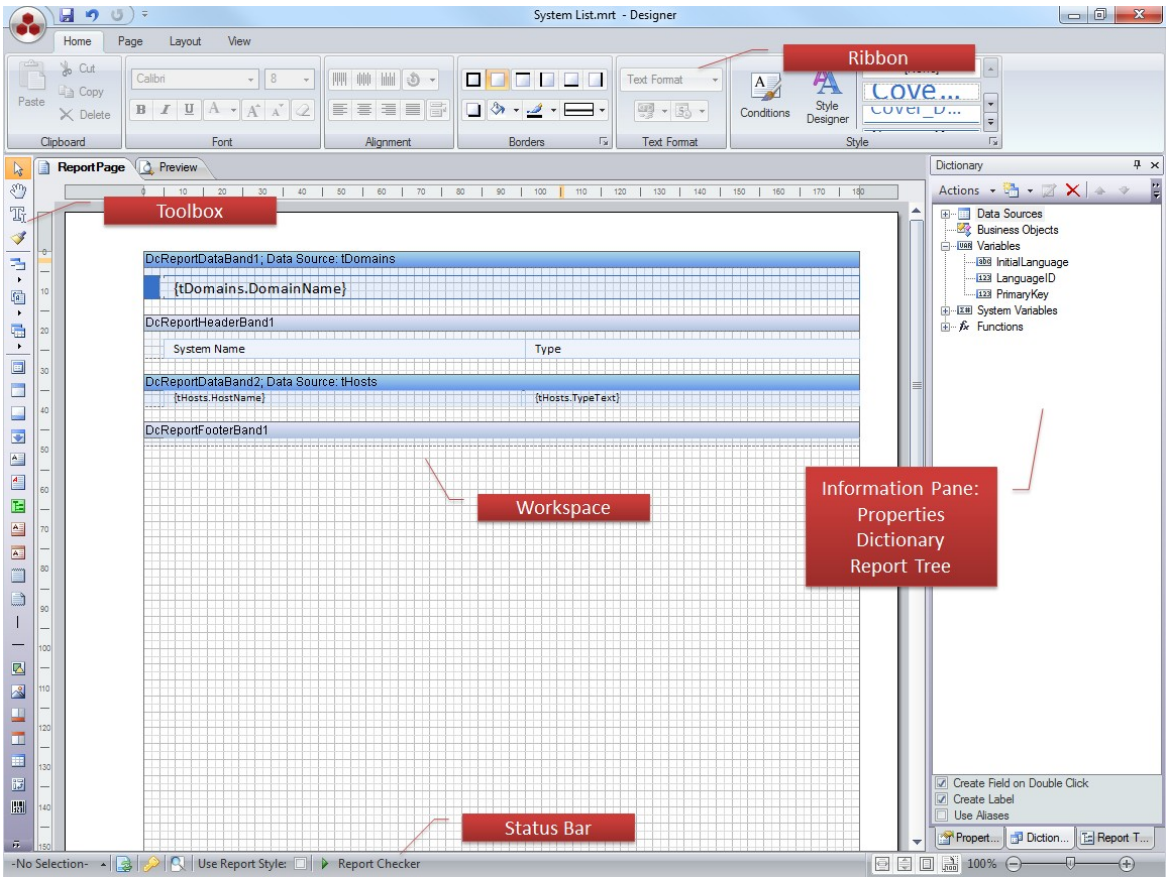
- The Report Designer can be opened from the *Manage Reports* dialog. To open the Designer module, the report to be edited must be selected. By clicking the *Designer* button, you can open the selected report in the Designer. When creating a new report, you first need to create it in the *Manage Reports* dialog. Then, you can open the empty report in the Designer.
- When you generate/execute a report from the tree view, an additional ribbon displays. This report can then be opened in the Report Designer by clicking the *Designer* button.

2.6.1 User Interface

Basic Structure

Use the Report Designer to create or edit reports.

The Designer consists of 5 main areas: ribbon, toolbox, workspace, information pane and status bar.



Ribbon

From the ribbon, you can define settings related to the layout and font design for the report.

- Home: On the *Home* tab, you can choose formatting options for the selected report component. These format settings can also be selected from the information pane. In addition, the *Cut*, *Copy*, *Paste* and *Delete* buttons are available from the *Clipboard* group. Conditions and styles can be defined from the *Style* group. By setting conditions, you can define a different format setting for values that reflect a certain condition.
  The Style Designer and the style selection field show the styles available in the current design.
- Page: On the *Page* tab, you can define the page size and other parameters. If your report includes a header, footer and cover page, these settings are not available.

The appearance of a report may also be changed after has been executed in Docusnap. In addition, you can add a watermark, either as a text or as an image, from this ribbon.

- Layout: On the *Layout* tab, you can specify how the components of your report will be aligned and how they will be stacked. This z-order can also be set by right-clicking the component and selecting one of the *Order* options. Using the *Size* button, you can resize multiple components to the same size. They will always take the size of the first component selected.
- View: On the *View* tab, you can select whether the components should align with the grid and which grid you want to display. Each data band has a header part used for identification. This header can be hidden. By clicking *Show Order*, you can display indicators that show the z-order of the components. The page can be displayed in either Normal or Page Break page mode. The tabs of the information pane can be shown or hidden. This is also true for the toolbox.

Toolbox

The main purpose of this toolbox is to make the components and various types of bands available when you create reports. To add a new component, you can simply click its icon in the toolbox and then click its intended position in the workspace.

Workspace

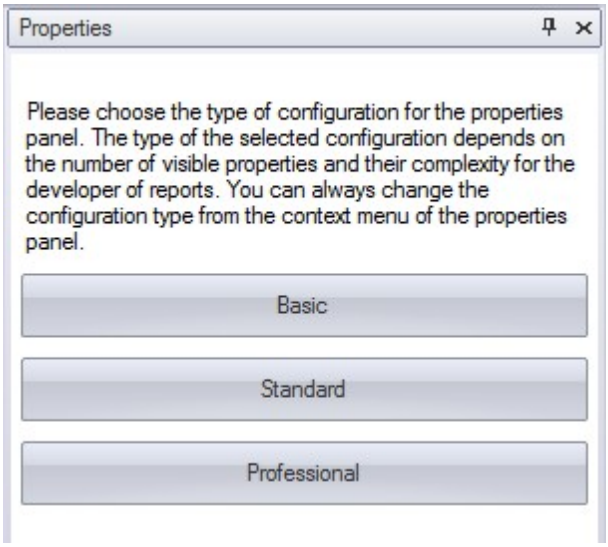
The report design workspace has two tabs. The first one displays the report page where you can create and edit your report. On this tab, you will create the data bands and define the connection to the database. On the *Preview* tab, you display a preview of the generated report. Since most predefined reports depend on the object they are linked with, you must specify the primary key before executing the report. If a primary key has not been provided yet, you can select it when changing to the *Preview* tab.

If the selected primary key is invalid or if no primary key has been provided, it is not possible to generate a data preview. Docusnap can only show the filtered data if you provide a valid value. Each time you change to the *Preview* tab, you will be prompted for the primary key until you have selected a valid key. If you would like to use a different primary key, you can change it immediately using the *PrimaryKey* variable on the Dictionary tab of the information pane. For this purpose, a corresponding icon is also available on the Report Designer status bar.

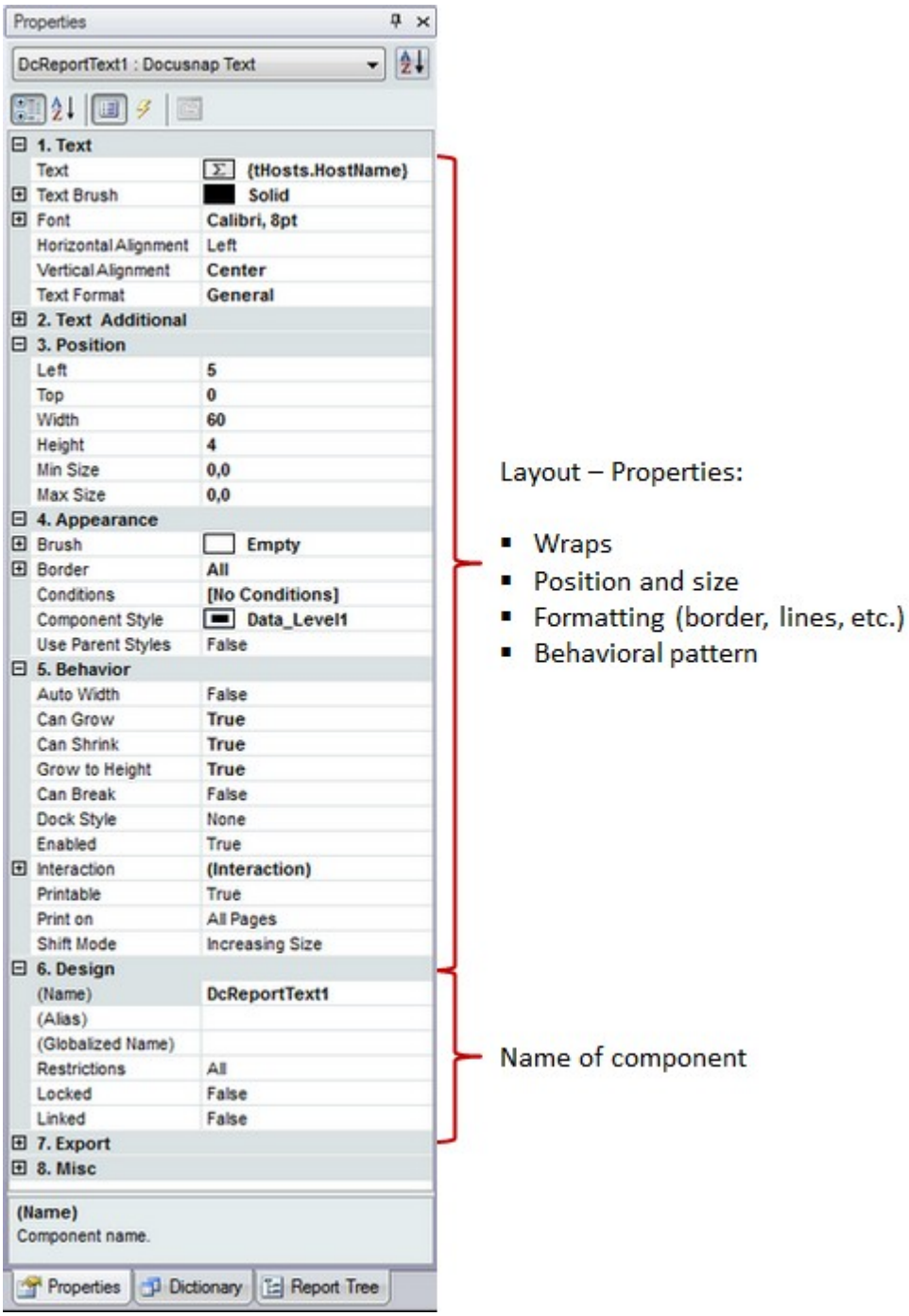
Information Pane

The information pane consists of three tabs.

- Properties: When you open the Report Designer for the first time, you need to specify one of the settings for the Properties tab. The available options are Basic, Standard and Professional. To be able to make detailed changes to the text boxes, charts, etc., select the Professional setting because it allows highest degree of modification. You can change this setting at any time using the context menu of the Properties tab.

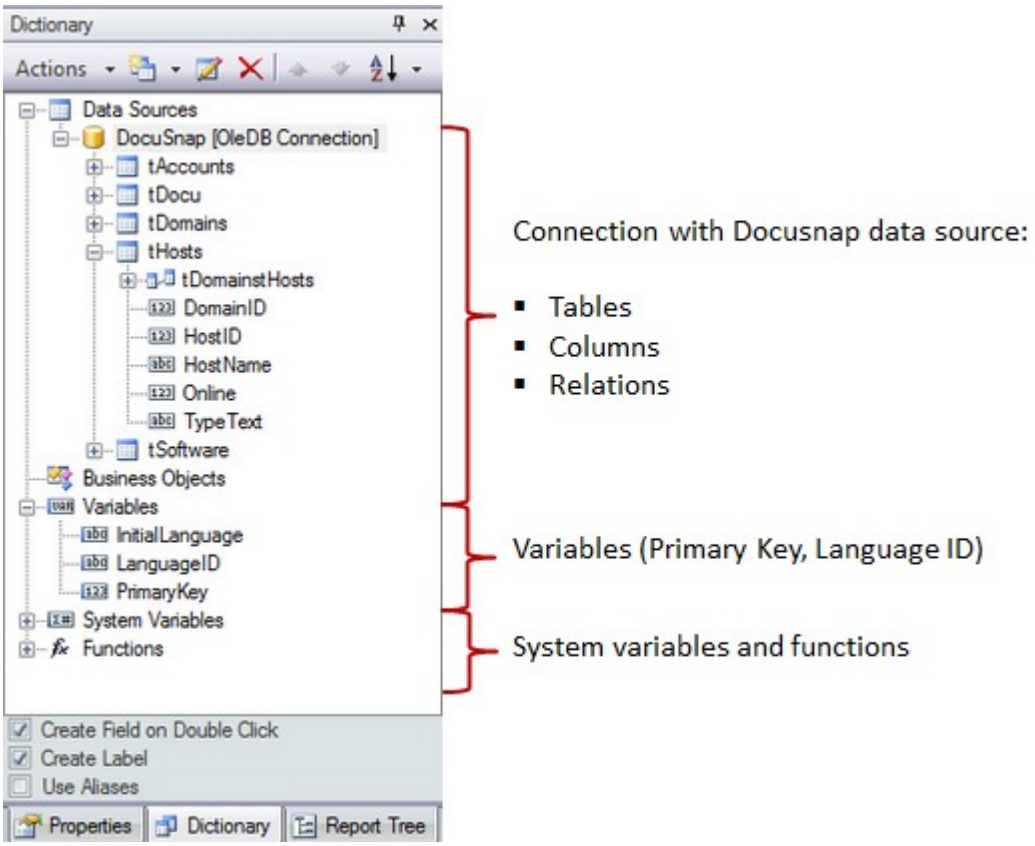


- The *Properties* tab displays the properties of the component selected in the workspace. Just like in the Designer for data entry screens, this tab enables you to specify the formatting, size, position and behavior of the selected component. A brief description of the selected property is displayed at the bottom of the information pane.



- Dictionary: On the *Dictionary* tab, the data source, variables, system variables and functions are displayed. The current Docusnap database will be used as the data source. All specified table definitions will be loaded. However, they are not yet connected to the database. Since only the tables needed for the current report will be connected to the database, the report will execute faster. To display the table that has been loaded into a data band, the *Connect on Start* property of that table must be set to *True*. With the options under *Actions* in Dictionary tab toolbar, you can save, open, add or create a dictionary. You can use the options under the button to establish a new database connection, create a data source or define a relation. In addition, new variables or categories can be added.

Click the button to open the selected object for editing. By clicking the button, you can delete the selected object. Clicking the Up or Down arrows will change the position of the selected item in the list. To sort the items alphabetically, click the button.



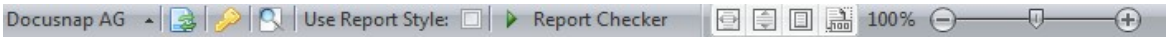
- Report Tree: This tab displays the components in a hierarchical structure. You can verify the organization of the individual components from the tree view.

Status Bar

- Company: The first button displays all companies created in the current database. You can define custom format settings for each company. When you select a company, Docusnap will apply the format settings for that company to the report. If -*No Selection*-displays, the format settings from the *Designs and Styles* dialog will be used.
- Update Styles: For each component, you can select the desired format settings on the *Properties* tab. If a different style template has been selected, the new format settings can be applied by clicking the button.
- Primary Key: Click the button to open a dialog where you can specify the primary key. After a primary key has been specified, data will be displayed on the Preview tab even if a filter has been applied to the primary key.
- Full Preview: When you click the icon, a full preview of the current report with

cover page, header and footer will be displayed. This is the same format as used when you generate the report in Docusnap from the tree view.

- Use Report Style: If you specify that the report style will be used for a report, this report will always use the style currently defined for it. If you change the style in the *Designs and Styles* or *Define Company Settings* dialog, the report will still use the original style.
- Report Checker: Checks the report for errors and displays them.
- Page width: Using the options to the right of the status bar, you can set the report page width and other parameters.



2.6.2 Basics

The purpose of the Report Designer is to create and edit reports. The datasheets and overviews that can be created in the IT Documentation module are based on reports and are also editable from the Report Designer.

Components

A report consists of various components. Primarily, data bands, text boxes and lines are used as components.

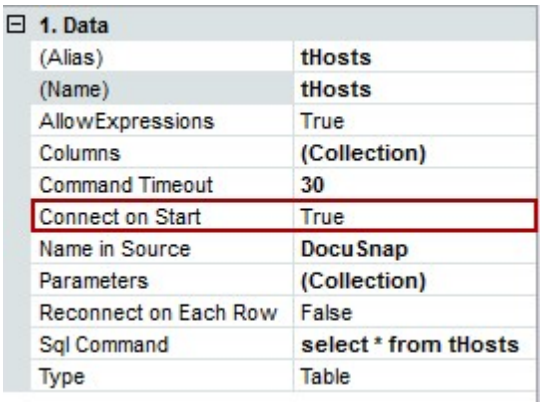
- Band: Each band is a placeholder for the data that will be shown at that position in the generated report. To show data from the database, you need to define data bands. Data filtering and sorting are set via the data band. The data band will be repeated as long as corresponding data is available in the database. In addition, data bands determine how the report will be organized.
  The header band is another useful data band. Using this type of band, column headings for the data can be shown. The header band will only be displayed in connection with a data band.
- Text boxes: Text boxes contain values from the database or static values.
- Line: Lines are used to separate individual values from each other, thereby creating a tabular display.

Data Sources

When you create a new report, all tables and relations are automatically created from the Docusnap database. The report is always connected to the database currently selected in Docusnap. Tables from the database as well as custom tables, which were created by means of SQL statements, may be used as data sources. In order to show the values from the data sources, you need to create data bands.

To reduce the time required for data loading, the tables are initially not connected to the database. To display the data from the table in the report, first select the

table to be connected from the *Dictionary* tab. Then, go to the *Properties* tab and set the *Connect on Start* property of the table to *true*.



1. Data	
(Alias)	tHosts
(Name)	tHosts
AllowExpressions	True
Columns	(Collection)
Command Timeout	30
Connect on Start	True
Name in Source	DocuSnap
Parameters	(Collection)
Reconnect on Each Row	False
Sql Command	select * from tHosts
Type	Table

It is also possible to enter SQL statements and thereby build a table that contains data from several other tables. To create a new data source, click the button or right-click the Dictionary tab and select *New Data Source*. The connection is always an OLE DB connection. Tables that are no longer needed can be deleted by clicking the button or by right-clicking and selecting *Delete* from the context menu. In addition to the tables, the relations between the tables are loaded into the Designer. You can also create new relations, if required. The columns that are used to define the relation must have the same data type.

Variables

When you create a report, three variables are created. These variables cause additional values from Docusnap to be added to the report.

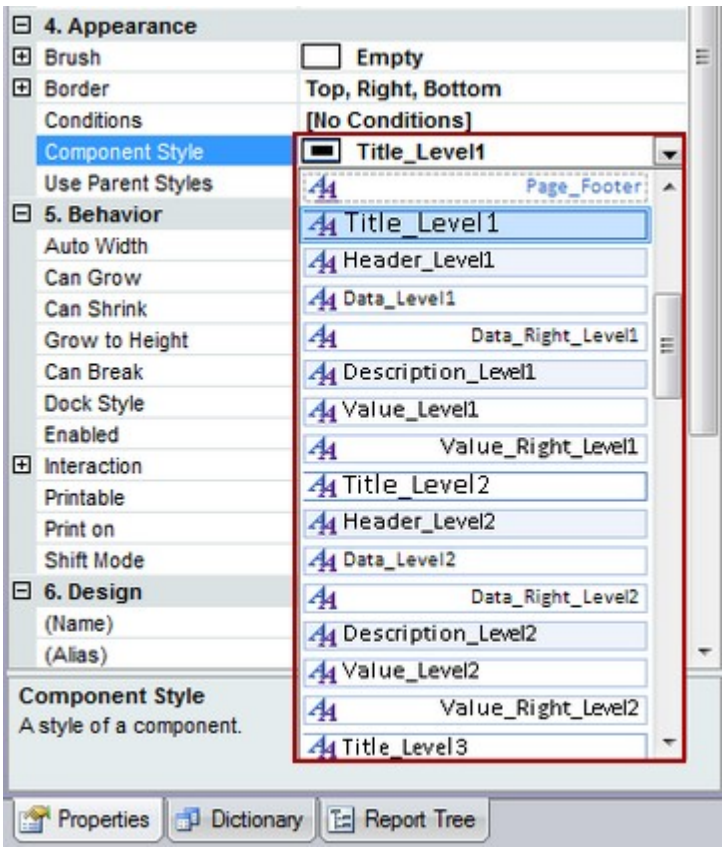
- PrimaryKey: Each report is linked with an object in the tree structure. Each object has a primary key which identifies the records of the underlying table. To make sure that a report will only include the data associated with that object, you can define a filter so that only records with a matching primary key will be used. This way, the report always shows the correct data for the domain, the computer, etc. When you generate/execute the report, the corresponding value will be assigned to the PrimaryKey variable.
  Since most predefined reports depend on the object that they are linked with, you need to specify the primary key. If no primary key has been specified yet, you can do so when changing to the *Preview* tab.
- LanguageID: The LanguageID for German is 0, and 1 for English. This refers to the language that was selected in the Manage Reports dialog. Using this variable, you can select the proper language for tables that exist in two languages.
- InitialLanguage: This variable can hold one of the following values: *TextEN* for English or *TextDE* for German.

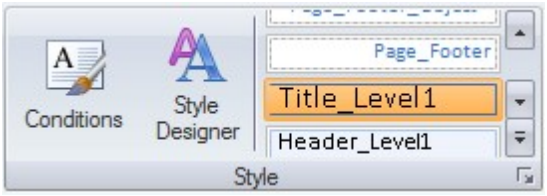
Preview

In the Report Designer, two types of previews are available. When you click the *Preview* tab, the report will be executed.
Clicking the *Full Preview* icon in the status bar will execute the report using the currently selected cover page, header and footer.
For most reports, the data will be filtered using a primary key. For this reason, you must specify an existing primary key to see data on the Preview tab.

Format Settings

- You can apply the desired style to each Docusnap component.
- To apply a style, open the *Properties* tab and click the *Component Style* property.
- As an alternative, you can use the button in the Style group of the Home ribbon to apply styles.
- Styles have been predefined for four different levels. The name of each style indicates its level, i.e. "Level 1" to "Level 4".

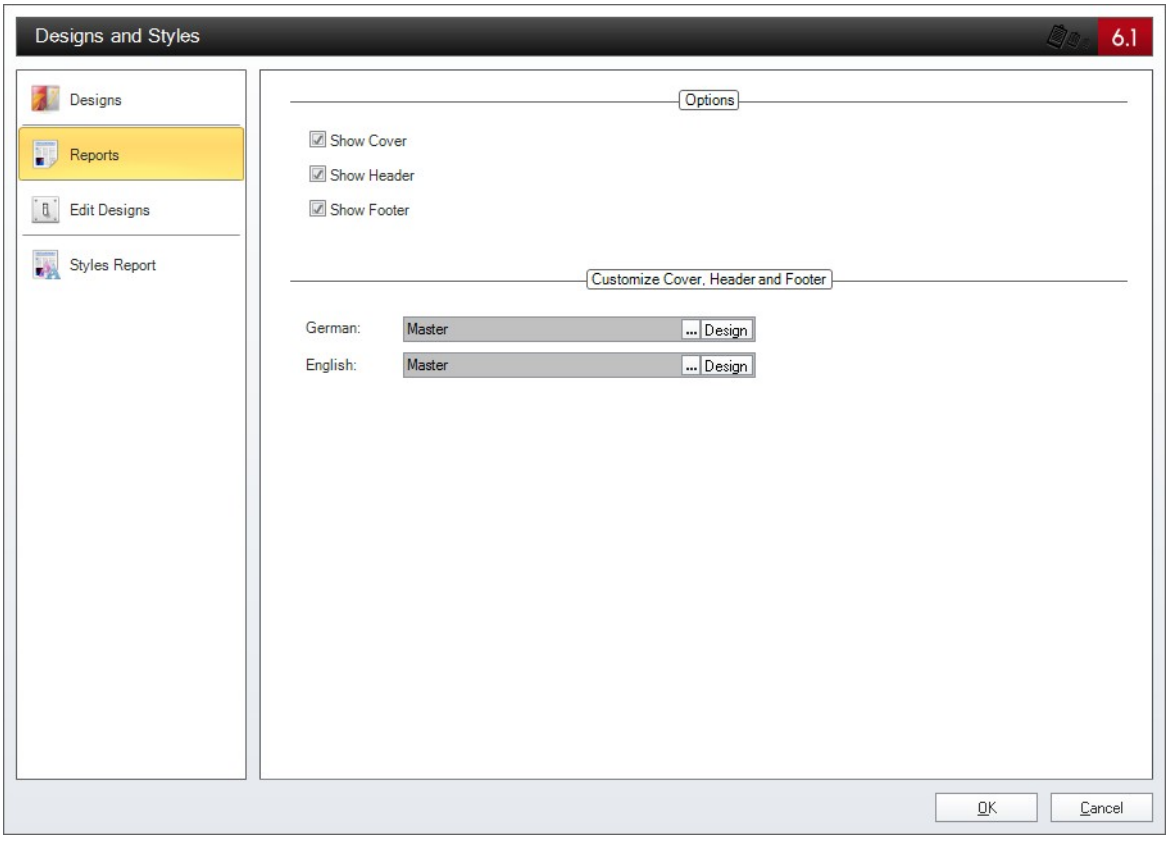




2.6.3 Editing Headers, Footers and Cover Pages

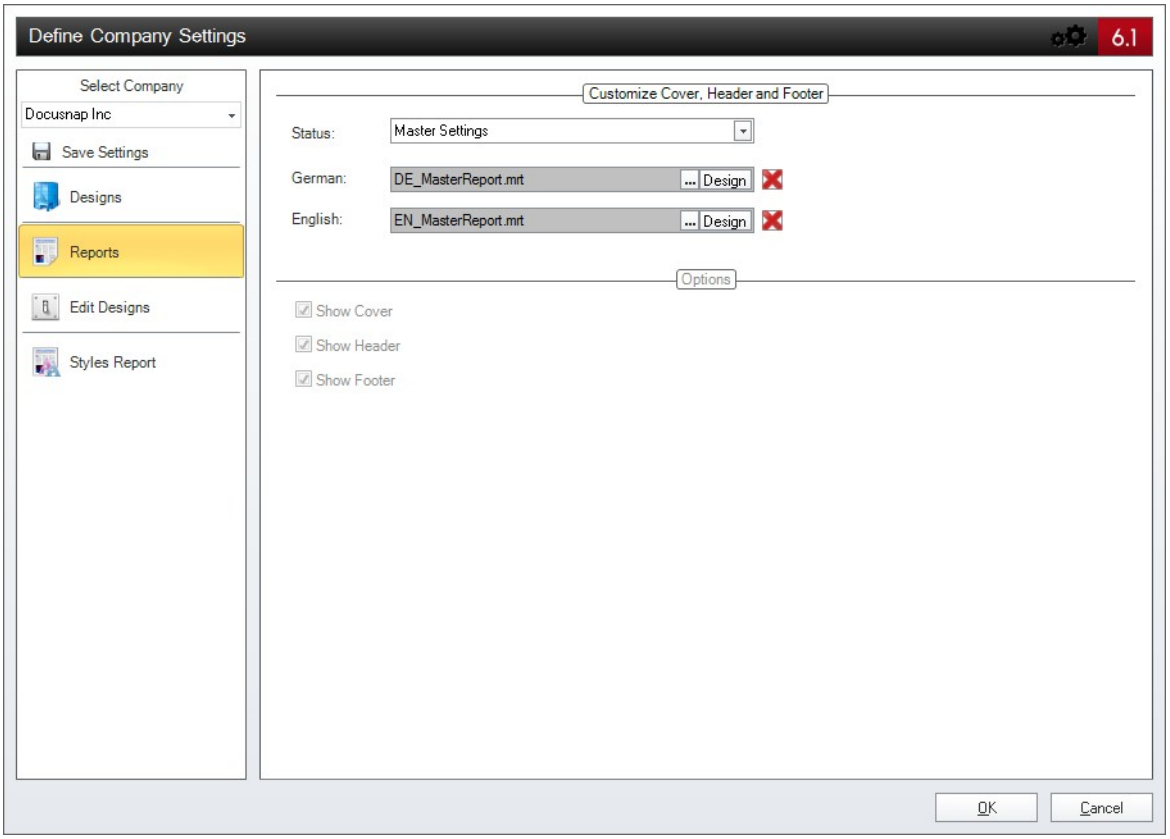
Docusnap provides default reports with predefined headers, footers and cover pages. In the *Designs and Styles* and *Define Company Settings* dialogs, you can edit the settings of these reports. To ensure that the header, footer and cover page will be used, either tick the appropriate checkbox(es) in the *Designs and Styles* dialog or configure individual company settings in the *Define Company Settings* dialog.

Open the master report in the Designer where you can edit the header, footer and/or cover page by clicking the *Design* button. A different report can be selected by clicking the ... button. The selected report will be saved in the report repository as *EN_MasterReport.mrt*. (For German templates, the "DE_" prefix will be used instead of "EN_".) Docusnap archives the file that previously had this file name and assigns it a timestamp so that you can restore it at any time through the operating system's file system.



In the *Define Company Settings* dialog, you can customize the header, footer and

cover page for each company. You can choose not to use the header, footer or cover page, to apply the master settings as defined in the *Designs and Styles* dialog or select an individual company-specific setting. Company-specific individual settings are always saved in the database. If you use the *Individual Settings* option, you can select an existing report and click the *Design* button to open it in the Report Designer and edit it. If no report has been selected and you click the *Design* button, an empty new report will be created which you can edit and save afterwards.



The EN_MasterReport.mrt file contains the header, footer and cover page definitions. Using the DsReportType property, you can specify the type of page, i.e. Cover, Header or Footer. When Docusnap creates the report, the width of the main report is compared with the width of the cover page, header and footer. If a cover page, header and/or footer of a suitable width exist for the report, these will be used. By default, the EN_MasterReport.mrt and DE_MasterReport.mrt files include a cover page, a header and a footer for reports in portrait and landscape formats, respectively.

The cover page will be output as the first page before the actual report.

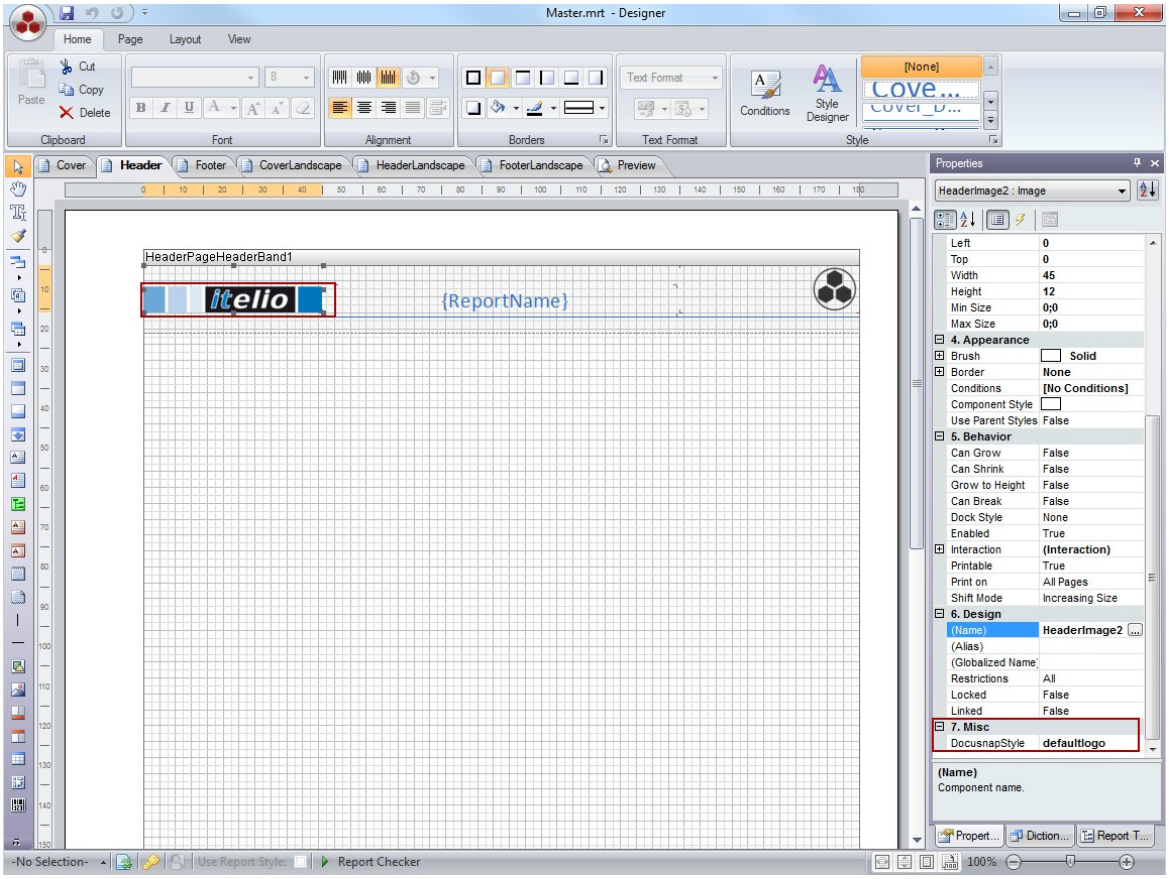
The header is output at the top of each page in the actual report. The header includes the report name and two logos.

The general logo (company logo of the documentation author) can be selected from the *Designs and Styles* dialog, the other logo from the *Define Company Settings*

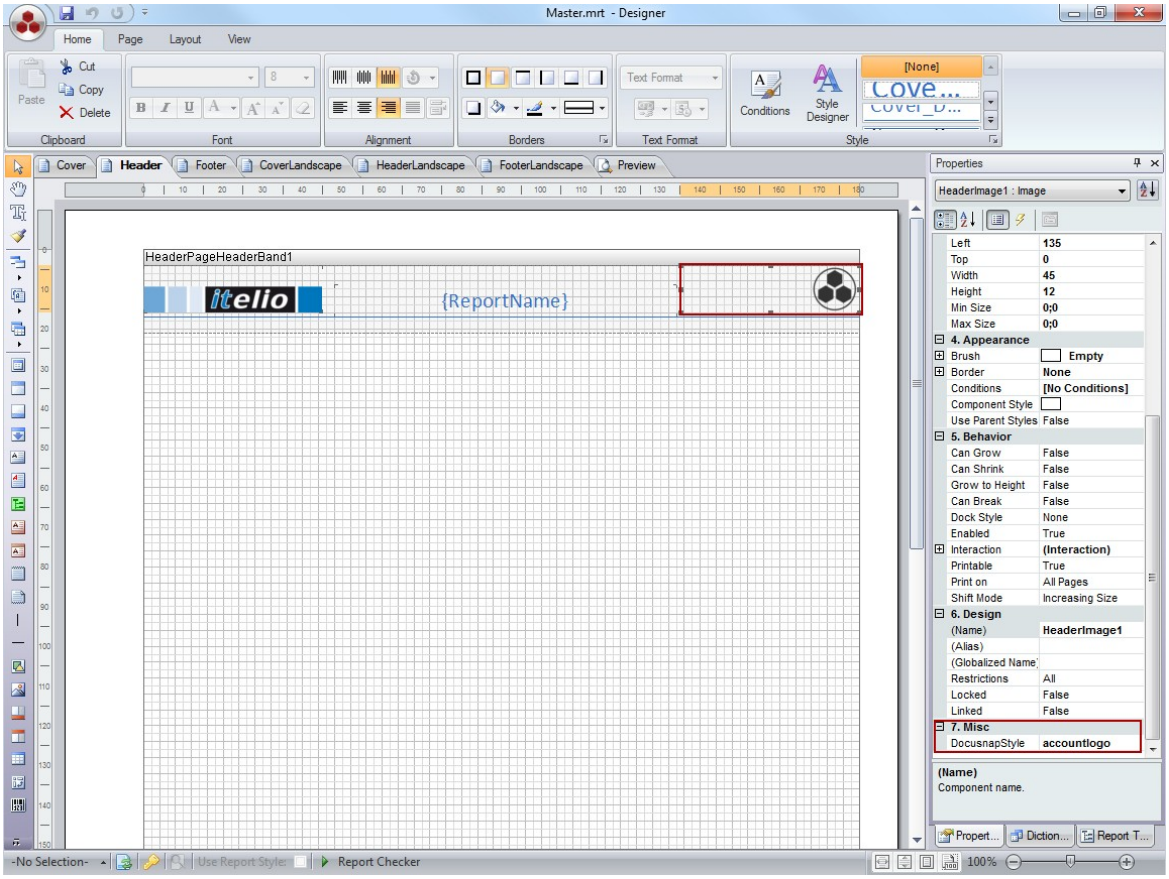
dialog. To change one of the logos, you do not have to open the header in the Designer. To change the logo displayed in the upper left corner of a report, select a different design in the *Designs* group of the *Designs and Styles* dialog. For each company you have created, a custom logo can be displayed in the upper right corner. You can select this logo from the *Define Company Settings* dialog.

To show the corresponding logo, a Docusnap image component is added in the Designer. On the Properties tab, you can set the *DocusnapStyle* property to either *defaultlogo* or *accountlogo*. This causes the display of the correct logo.

- defaultlogo: If you set the DocusnapStyle property of an image to *defaultlogo*, the logo selected in the *Designs and Styles* dialog will display.



- accountlogo: If you set the DocusnapStyle property of an image to *accountlogo*, the logo selected for the current company will be displayed.

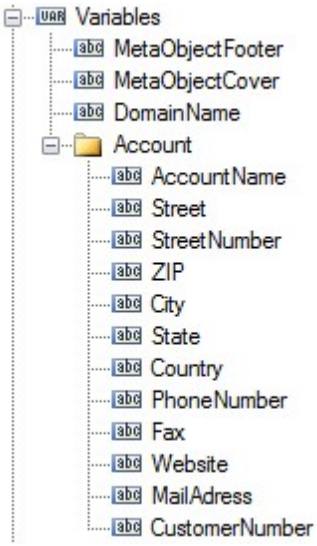


When creating a new text box, image, line, or data band component, make sure that its name is different from the names of the components in the header, footer or cover page. This is also true for the rest of the report: The name of each component in the report must be unique. If two components have the same name, only one of them will be shown in the generated report. For this reason, when creating an additional component for the cover page, header or footer, prefix its name with *Cover*, *Header* or *Footer*. The component name can be changed from the Properties tab.

- Variables for the Current Meta Object
  - MetaObjectCover: This variable shows the name of the object this report is linked with. It is associated with the cover page.
  - MetaObjectCover: This variable shows the name of the object this report is linked with. It is associated with the footer.
  - DomainName: This variable shows the name of the domain under which the report will be generated. If the report is generated above the domain level, this variable is blank.
  - Account: To show information on the company for which the report will be generated, several variables are available. For example, you can use the

AccountName variable to include the company name in the report.

- CoverAccountID, HeaderAccountID and FooterAccountID: These variables include the primary key of the company for which the report will be executed. This way, you can filter the database on the company name. The respective variables are used for the cover page (CoverAccountID), the header (HeaderAccountID) and the footer (FooterAccountID). When the report is generated, these variables will be replaced with the actual data. Since separate company-related variables are now available in Docusnap 6.1, the variables mentioned above are no longer required in the default MasterReport.

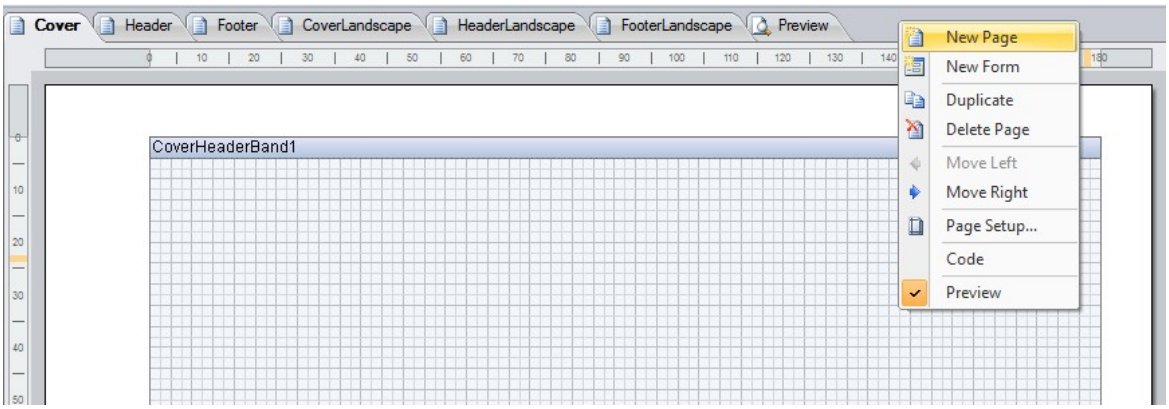


- System Variables
  - ReportName: Docusnap uses the report name specified in the *Manage Reports* dialog.
  - ReportDescription: This is the description stored via the *Manage Reports* dialog. If no description was entered, no description will be displayed.
  - Today: The *Today* variable shows the current date.
  - ReportAuthor: This is the author as specified in the *Manage Reports* dialog. For the predefined Docusnap reports, the author is "Docusnap". To display the name of a different author, you first need to change the value in the Author field of the *Manage Reports* dialog.
  - TotalPageCountThrough: This variable shows the page count for this report.
  - PageNofM: This variable shows the current page number and the total page count.

Additional Page Formats

By default, each report includes a cover page, a header and a footer for the A4

portrait and landscape formats with right and left page margins of 2 cm and top and bottom margins of 1 cm, respectively. If required, you can define a cover page, header and footer for a different page format. To do so, right-click the area next to the pages tabs to open the context menu and select New Page.



For this page, set the DsReportType property to Cover, Header or Footer, as required. Then, assign the desired page format to the new page. You can now create the content to be used later for the cover page, header or footer. If you later create a report with the same page width as the new pages, the newly created cover page, header and footer will be used.

Make sure that the page names are unique, otherwise the report cannot be displayed. If the MasterReport includes multiple pages with the same page width, the leftmost page will be used first.

2.7 Report Creation

This section uses a sample report to explain how you can create reports.

You will learn how to create a report yourself. In addition, this section explains the use of relations to specify which data will be shown in the report.

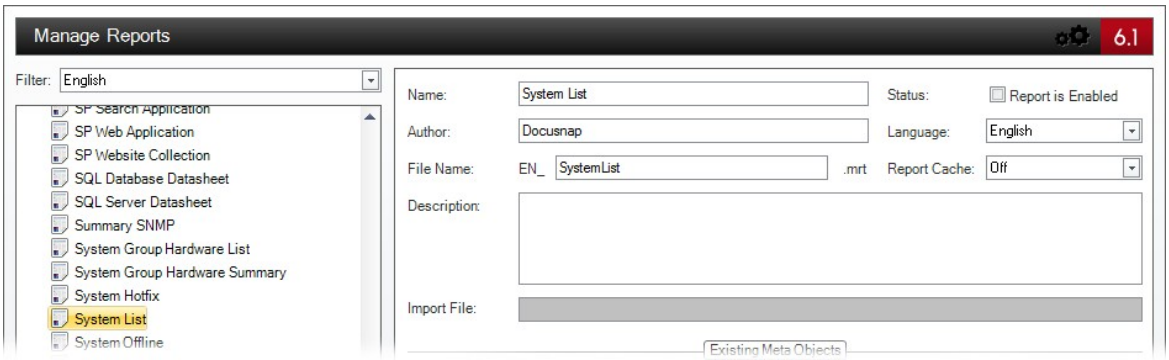
In an example, you can see how styles are applied to a report.

2.7.1 Initial Steps

The initial steps for creating a report will be explained using the example of a report that lists all computers.

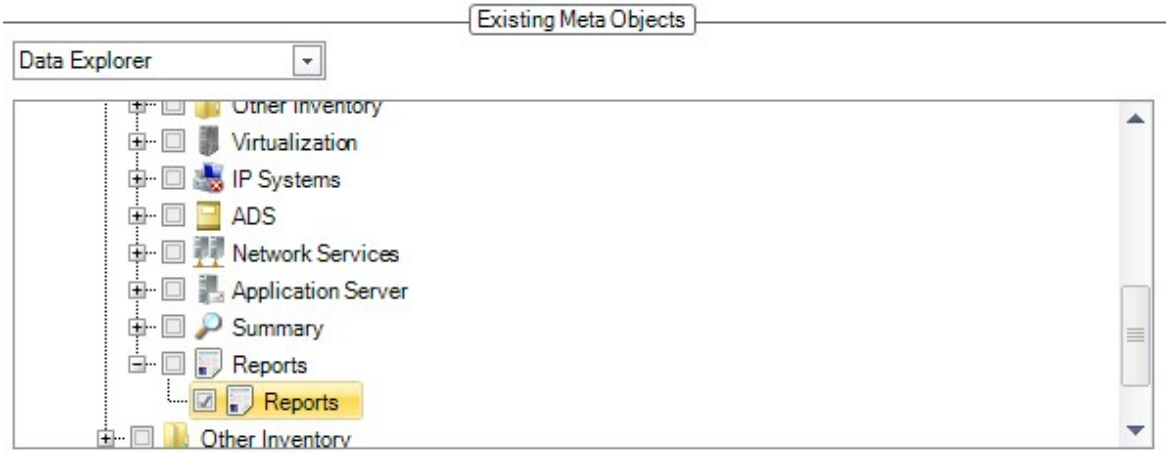
Creating the Report

To create a new report, go to the *Manage Reports* dialog. A new report can be created by clicking the *New* button. Then, specify the required properties. These properties will be displayed on the cover page.

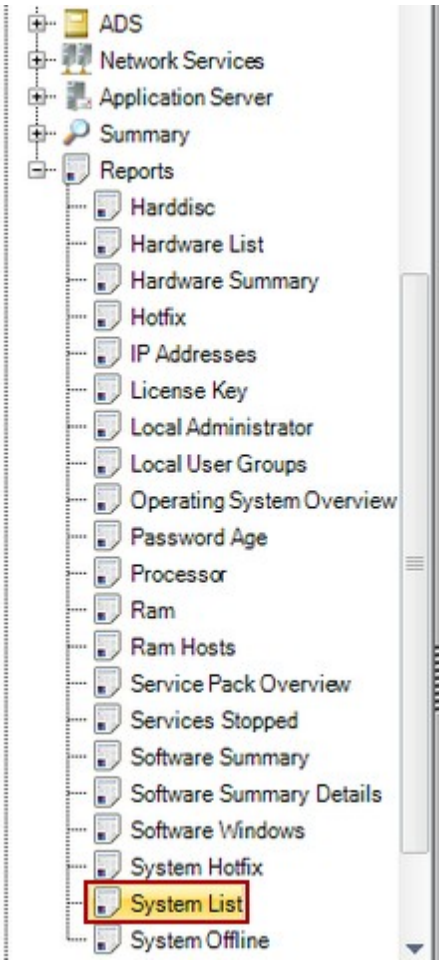


Linking the Report with a Meta Object

In the next step, you need to select the location in the tree view where the report will be executed. In this case, select the *Reports* meta object below the domain level. This passes the primary key of the domain to the report, and the report will be displayed below the domain level.

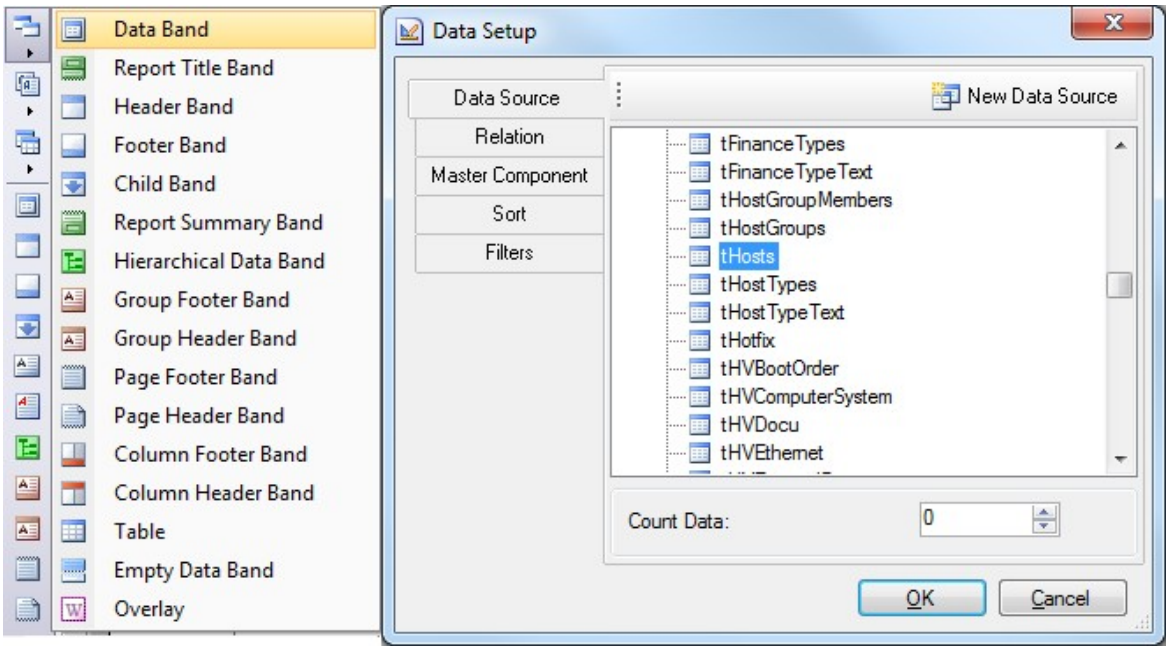


After it has been saved, the report will be displayed in the tree view. To generate/ execute this report, click it. This opens the additional *Reporting* ribbon. The report can be opened in the Report Designer by clicking the *Report Designer* button on this ribbon.



Creating a Data Band

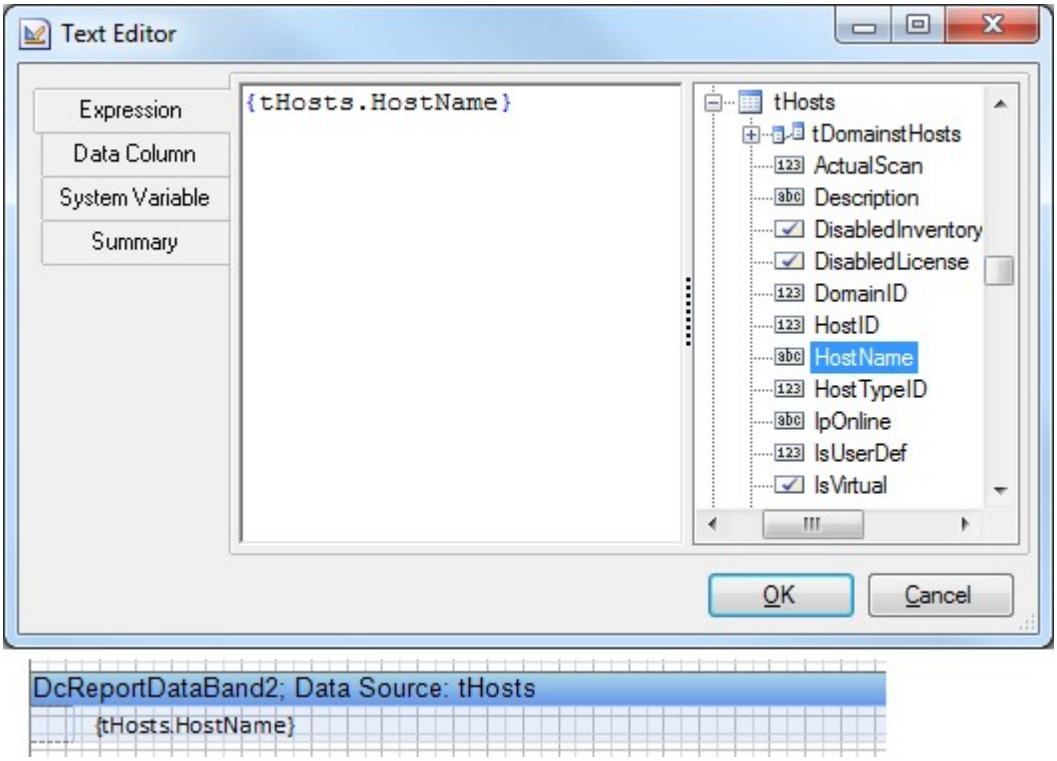
A data band is required for the output of data from the database. Data bands can be added from the *Docusnap Bands* icon in the Report Designer toolbox. To add a data band, click the *Data Band* component. For the data source, select the tHosts table.



Adding a Text Box

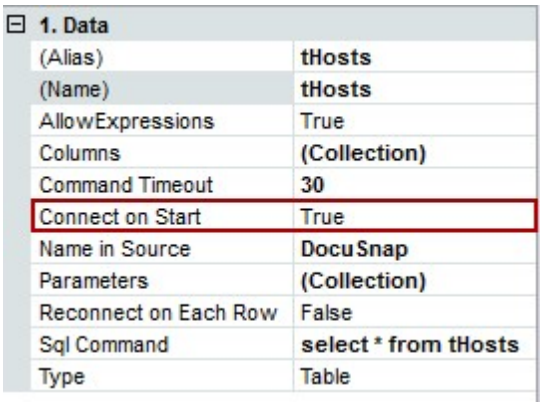
Text boxes are required to enter values that will be shown in the report. You can combine expressions and table columns in a text box. Text boxes can also be added from the toolbox (Text components).

Draw the text box in the data band. In the Text Editor, select the *HostName* column from the *tHosts* table.



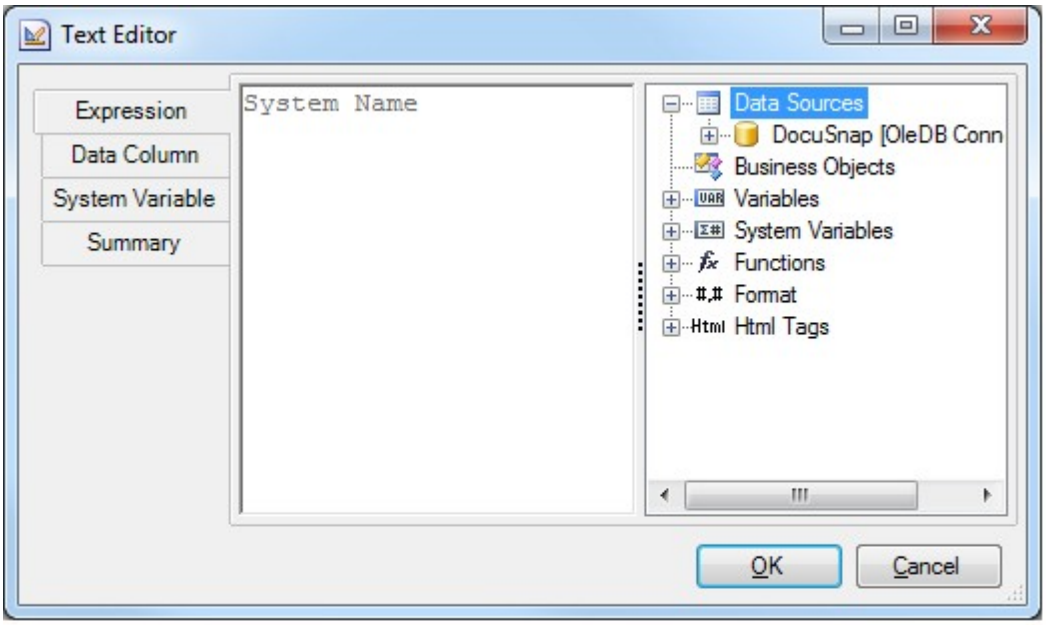
Connecting the Table

Before the table data can be displayed in the report, you need to connect the table to it. For this purpose, set the *Connect On Start* property of the table to *True*.



Creating a Header Band

The header contains the headings for the data. The header band will only be displayed if the data band contains data. First, add a *Header Band* component from the toolbox. Use drag and drop to place the header band before the data band. Then, add a text box and enter the heading.



Primary Key

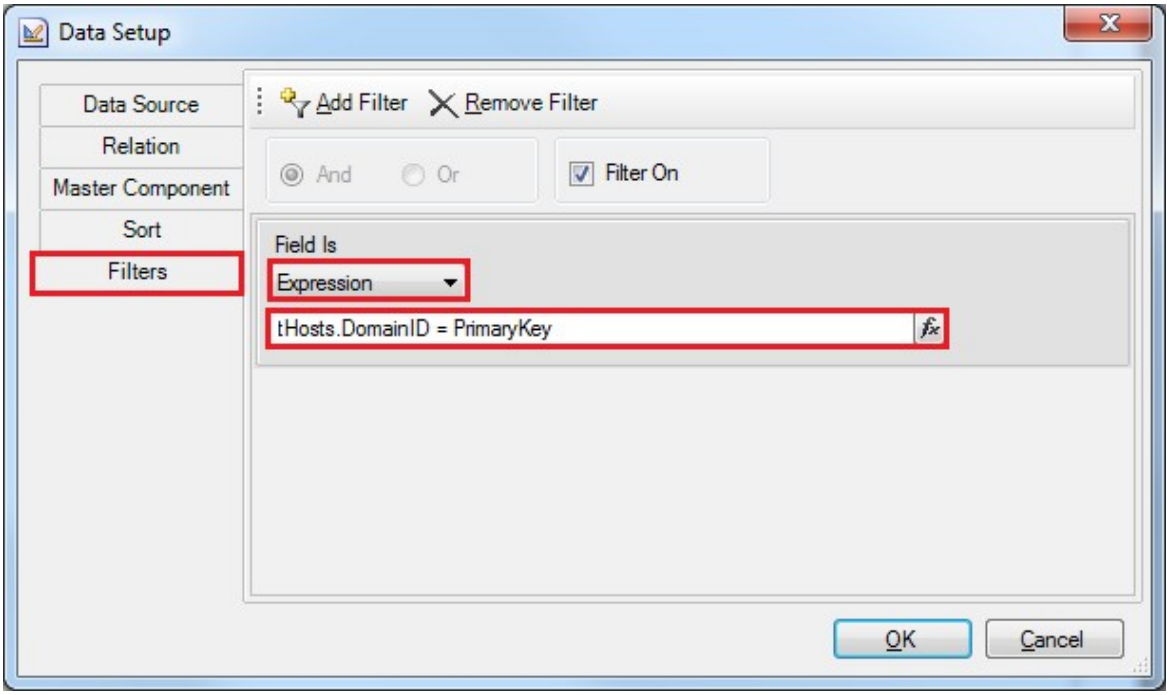
The primary key is stored in the PrimaryKey variable. The primary key of the associated meta object is used as the primary key. By means of this value, the data for the report can be filtered on the associated object. This means that only data is included in the report that is associated with that computer, domain, license group,

etc. The variable will be set when you execute the report.

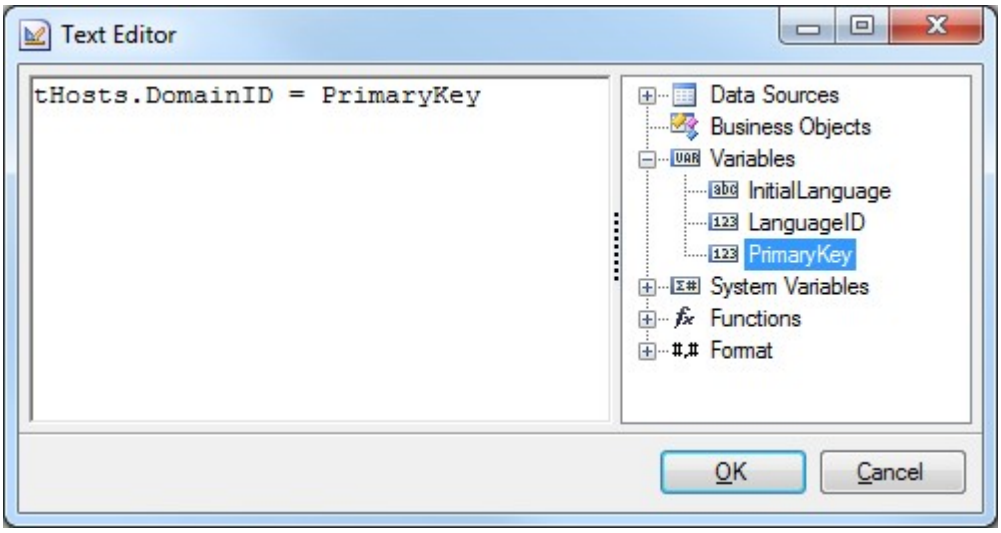
Filtering on the primary key can be done in two ways, either via the data band or directly using an SQL statement.

Specifying the Primary Key in the Data Band

You can define a filter for a data band. Open the dialog by double-clicking the data band. To define a filter, open the *Filters* page and click *Add Filter*. Then, change the option for *Field Is* from "Value" to "Expression". A click on the button will open the text editor. Enter the filter condition in this dialog. It is also possible to enter the filter condition directly in the expression field of the *Data Setup* dialog. The advantage of the text editor is that you can add the column names by double-clicking them.



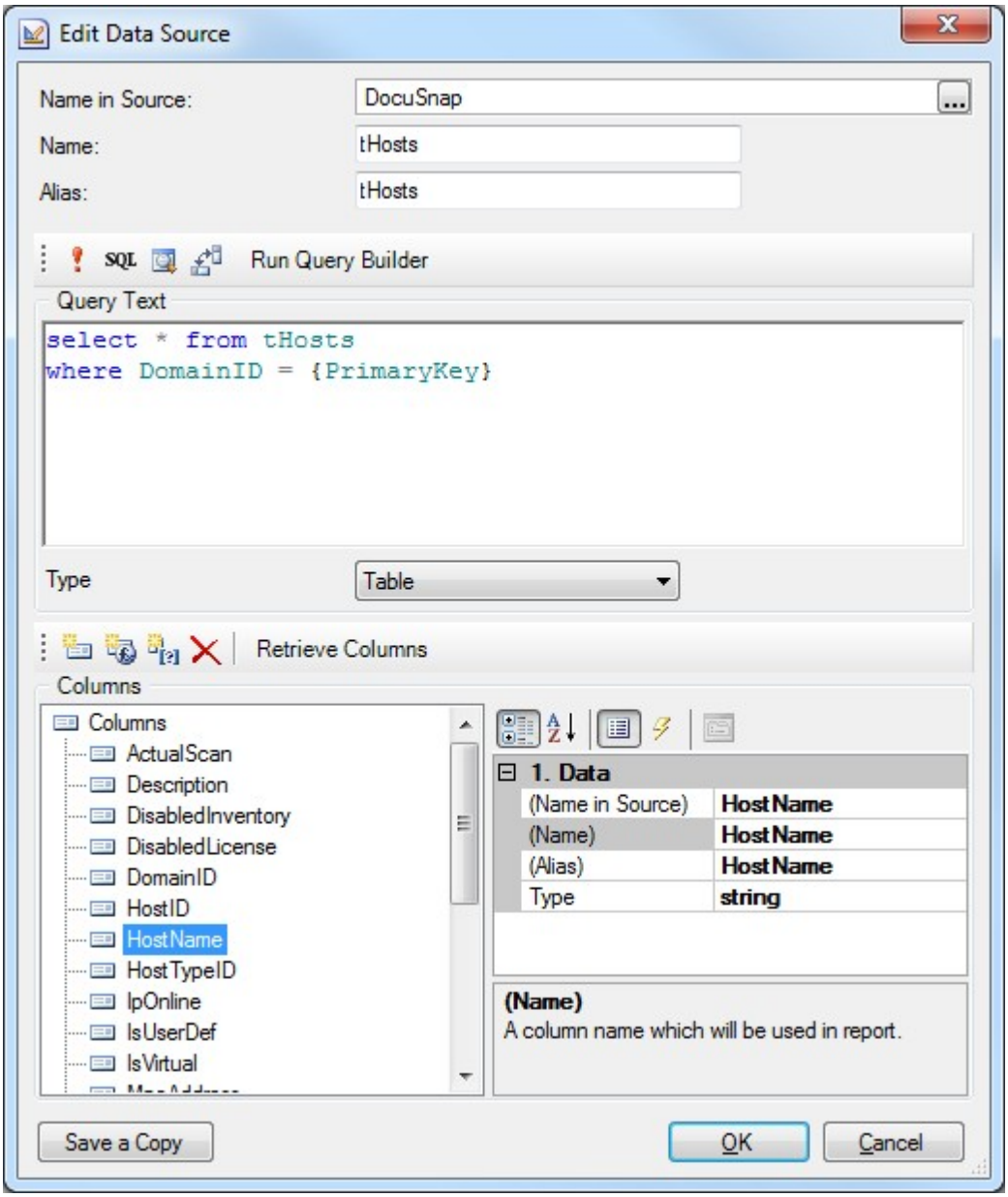
To the right of the text editor, the contents of the dictionary are displayed. You can add the desired column by double-clicking it or by using drag & drop. Then, enter an equals sign ("="). Next, you need the PrimaryKey variable. You can find it under the *Variables* node. Add it by double-clicking or using drag & drop again. Confirm your filter settings by clicking *OK*.



Specifying the Primary Key Using an SQL Statement

Filtering on the primary key can also be done by specifying an SQL statement for the table. Only data that corresponds to the primary key will be included in the report.

To edit the data source, go to the Dictionary tab, select the table, right-click and select *Edit*. Then, you can create an SQL statement with a filter on the primary key using a *Where* clause. The result is that only data that matches this primary key will be shown in the report.



Primary Key: Filter vs. SQL Statement

The advantage of the SQL statement over a filter will become apparent in the speed of execution of the report when large data sets need to be accessed. When you use a filter, all data in the table will be retrieved from the database. Then, Docusnap determines the data output by means of the filter. If you filter the database using an SQL statement, only the data that matches this SQL statement will be retrieved. This reduces the execution time of the report as less data must be loaded.

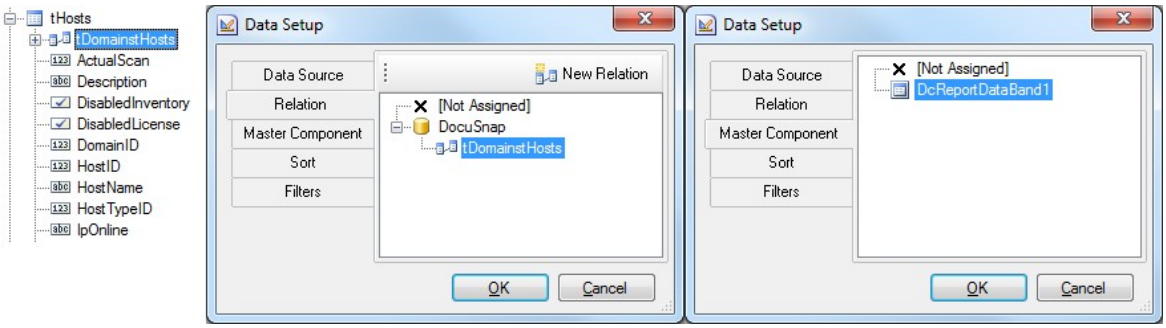
2.7.2 Components and Functions

Before continuing with the creation of the report, you will now be introduced to working with data bands and text boxes. In addition, a brief explanation of the functions that are available for reports will be provided.

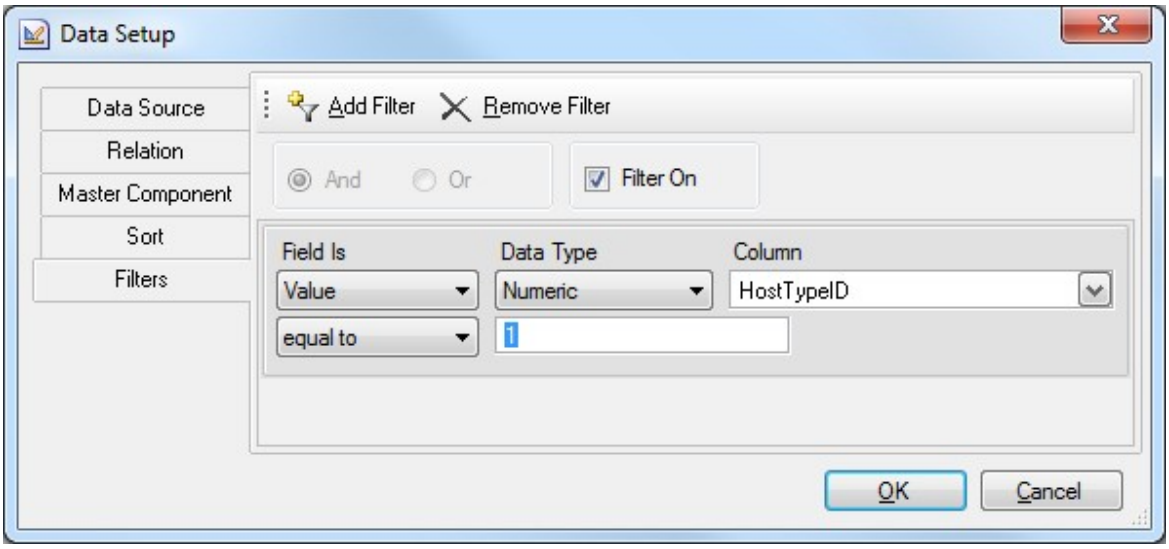
Bands

Bands are key components of a report. Via data bands, you establish the connection to the database. In addition, they can be used to sort and filter the data. Data bands can be added from the toolbox. They fulfill a variety of tasks. Each band has a different color to make them more distinctive.

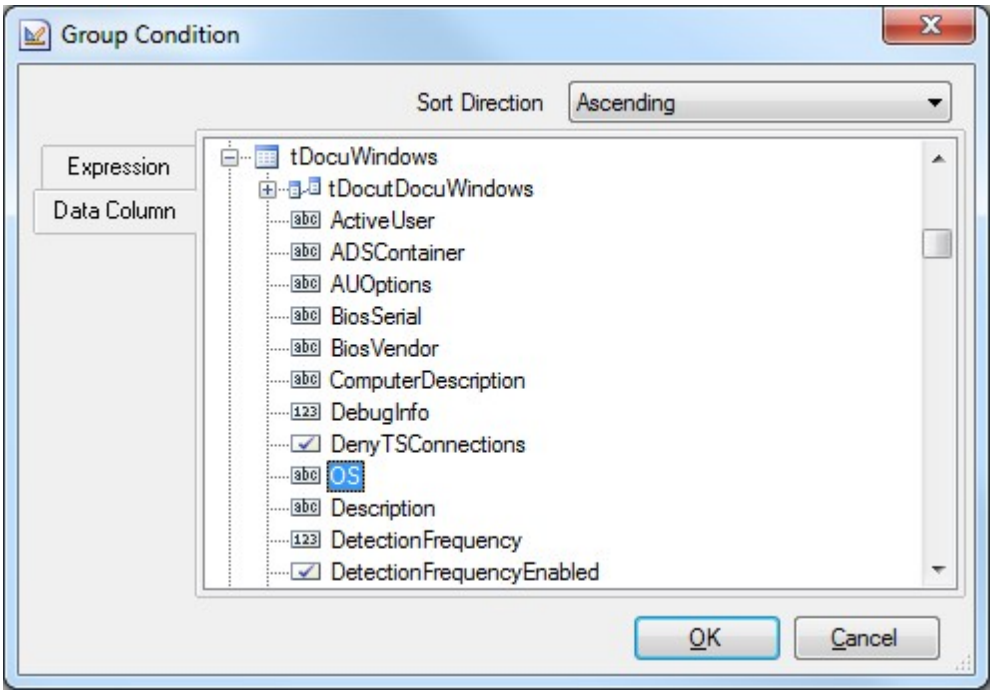
- Data band: The data band connects your report to the database. When you create a data band, the dialog for selecting the data source, relation, master component, sorting, and filtering conditions opens. From this dialog, you can select the data source. The data band is repeated as often as corresponding data is available in the table. You can restrict the amount of data to be included in the report by specifying relations and filters.
  - For a relation, you need to select a master component. This ensures that only data that matches this master component will be shown. A relation must exist between the tables selected for the data band and for the master component. When you create a new data band or double-click an existing data band, a dialog opens. From this dialog, you can select the relation and the master component. The report shows the first record from the master component and then immediately below it, the associated records from the sub-component. Then, the next record from the master component is shown and again the associated records from the sub-component. This pattern is repeated until there are no more master component records. Example: The domains that have been inventoried can be assigned to the company accounts.



  - Filters are used to filter the data on one or more values. When you create a new data band or double-click an existing data band, a dialog opens. From this dialog, you can define the filter. If required, you can specify multiple filters that can be combined using either *And* or *Or* operator.



- Header Band: The header band, the page header band and the report title band can be used to show headings and titles/headings at the desired position in the report. The header band provides the headings for a specific data band. The header band is shown before the data of the associated data band. If the data band is empty, the header band will not be displayed either. The page header band is shown on every page, and the report title band is shown only once, i.e. at the beginning of the report.
- Footer Band: The footer band, the page footer band and the report summary band are the closing items of the associated data display. The footer band is shown after the data of the data band. The page footer band appears at the bottom of each page, and the report summary band at the end of the report.
- Group Header Band: Using a group header band, you can group data of a table. For instance, the data can be grouped by operating systems. When you add a group header band, the dialog for entering the group condition opens. Here, you can select the column that controls the grouping and you can specify the desired sort order for the data. Then, a separate data band must be defined for the remaining data. In the heading, the value selected in the condition will be shown and the data will be grouped accordingly. Make sure to add a data band after the group header band. Otherwise, the group header band will not be displayed either.



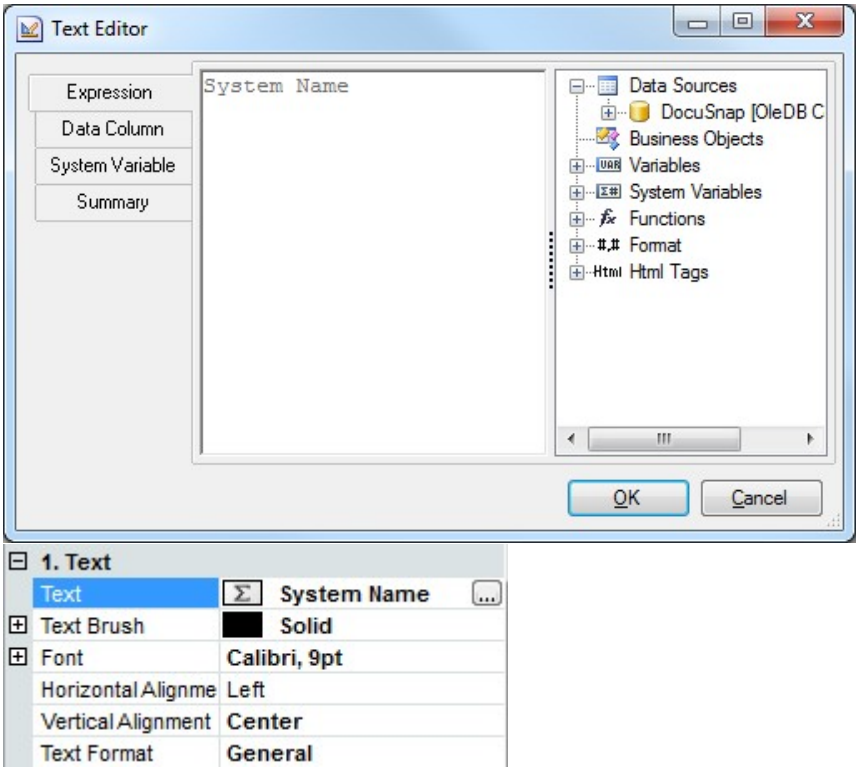
- Group Footer Band: This band can, e.g., be used to draw a line below the group. However, it is also possible to calculate the number of records.
- Column Header Band: The data in the data bands can be shown in columns. The column header band is automatically divided into as many columns as the associated data band has.
- Column Footer Band: This band is the closing item of a data band consisting of multiple columns.
- Hierarchical Data Band: This band enables you to display a recursive table. A recursive table has a column that depends on another column in the same table, for example, a list of employees where one employee may be the manager of another. The properties for the hierarchical data band include a group of *Hierarchical* properties. In the *Key Data Column* field, you can specify the column that contains the key for the data. In the *Master Key Data Column* field, you can specify the column to which the hierarchy refers. The *Parent Value* field allows you to enter a value that indicates the top of the hierarchy.
- Child Band: This band is an extension of its parent band. Even though it is possible to place multiple data fields from the tables in a single data band, the size of the data band does not adjust as desired when you change the font size. For this reason, child bands are used. The color of the child band is slightly lighter than the color of its parent data band.

Text Boxes

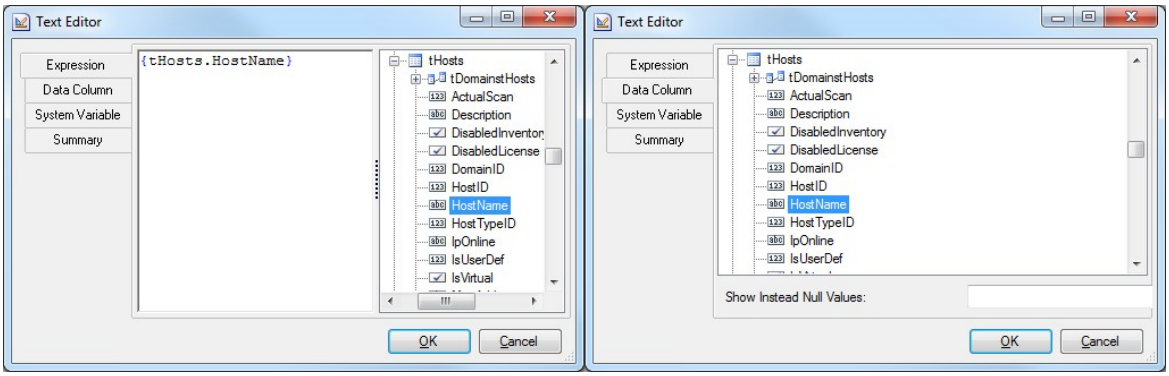
Text boxes can be used to show the headings and data of your report. When you add a text box, the Text Editor opens. In the Text Editor, you can either add fields

from the database or enter text.

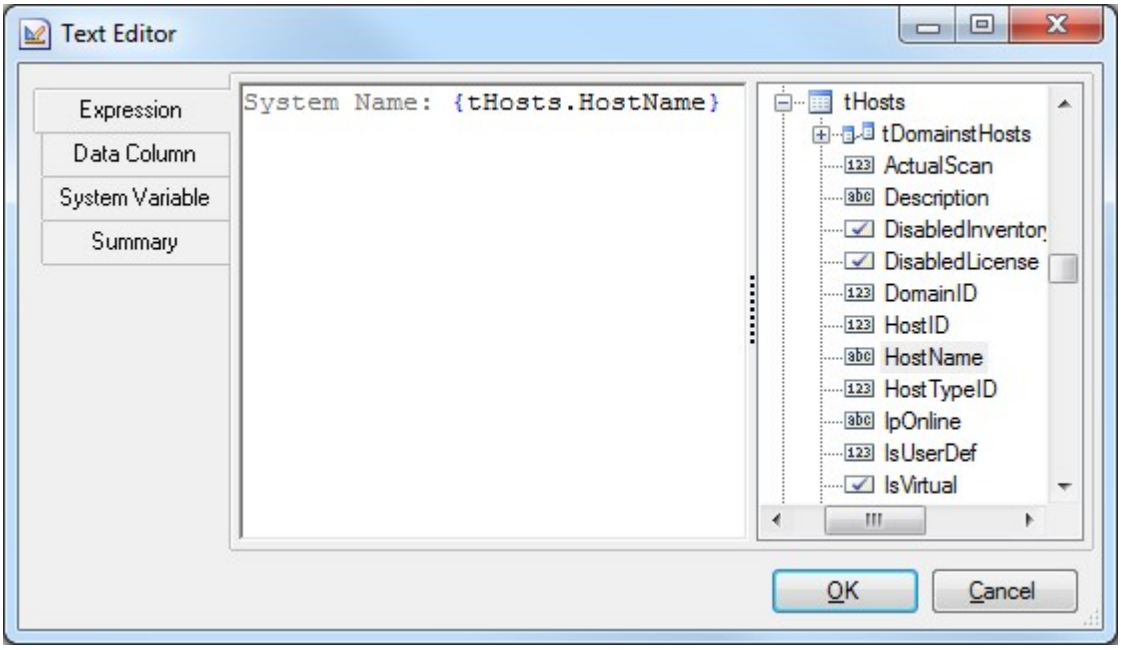
- Heading: The heading for the data is text that has been entered manually. You can enter this text in the Text Editor. To apply the text to the text box, click the *OK* button. Alternatively, open the *Properties* tab where you can enter the text in the *Text* field. To open the Text Editor, either click the ... icon or double-click the text box.



- Data: If you want to output data in a text box, place the text box on a data band, a group header band, a hierarchical data band or a child band belonging to another data band. In addition, the data source for the band must match the table associated with the column. Otherwise, there is no connection to the database and the data from the database cannot be shown. In the Text Editor, you can select the data fields either on the *Expression* or the *Data Column* page. To add a field from the *Expression* page, double-click it. On the *Data Column* page, click the desired field. Then, click the *OK* button to apply the selected data field to the text box.



- Expression: In text boxes, data fields and text may be combined. In this case, you can only use the *Expression* page. When using the Expression page, you can enter a name that will precede the data column. Make sure to place such a text box on a data band, a group header band, a hierarchical data band or a child band belonging to a data band. Furthermore, the data source for the band must match the table associated with the column. Otherwise, the data from the database cannot be shown, because there is no connection to the database.

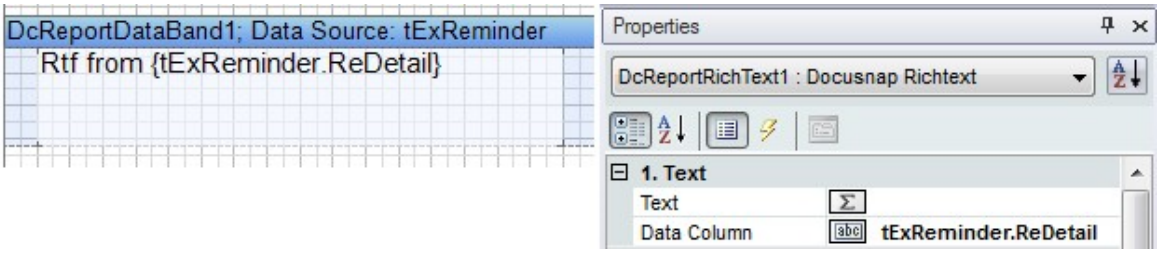


- Functions: In addition to data fields and text, you can specify functions to be included in a text box. The functions available in Docusnap will be explained in the Functions section.

Rich Text

For data entry screens, it is possible to create rich text boxes. In order to show the format settings of these fields in the report as well, you need to define a corresponding rich text box in the report. Rich text boxes can be moved and handled just like text boxes. The difference between a rich text box and a normal text box is that the data column to be included in the report will not be entered in

the field, but rather specified from the Properties tab. You can select the data column by setting the *Data Column* property on the Properties tab. The text will only be displayed with the selected format settings if you specify the data column through this property.

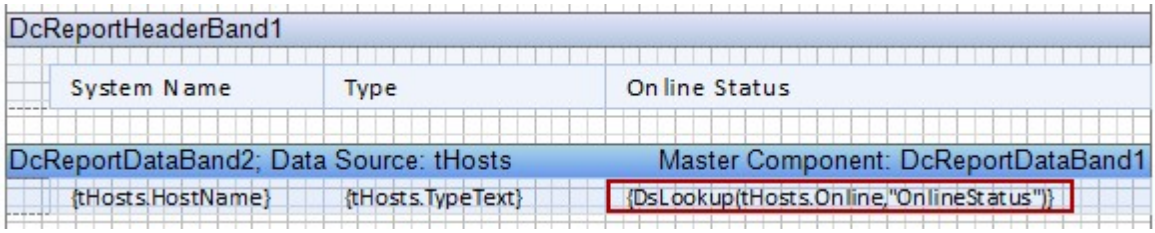


Functions

Docusnap provides several functions for reports.

General Functions

- DsLookup: Many tables contain only numbers. To express these numbers as words, the tSysInitials table from the Docusnap database can be used. In this table, the states of services and computers, the computer types, the countries, and other information are stored as literals. The InitialFilter column contains the values stored in the respective tables. The DsLookup function can be used to show the values from these tables as literals. Enter this function into a text box located in a data band. The function must be surrounded by braces "{}", otherwise the Designer will not recognize it as a function. Thus, the form of the function is: `{DsLookup(ColumnName,"InitialFilter")}`. The column name represents the value in the table that should be replaced with the literal. "InitialFilter" is the word that exists in the InitialFilter column of the tSysInitials table. Thus, if you want to display the online status for the tHosts table, enter the following function: `{DsLookup(tHosts.Online,"OnlineStatus")}`. For an overview of the reference values, refer to the Reference Values section in the appendix.



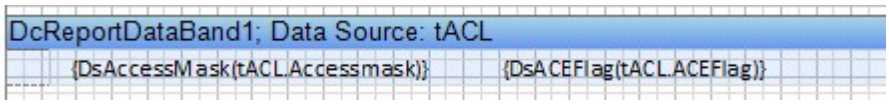
- DsSortIPAddress: If you wish to sort a data band by IP addresses, the number blocks with less than three digits need to be left-padded with zeros. This means that if you want to sort the data in a data band by IP addresses, precede the name of the IP address column with the `DsSortIPAddress(ColumnName)` function. This function is used by the *IP Addresses*report.
- DsGetDateFromTimeStamp: The date from certain Active Directory properties, such as the creation date, is stored as a timestamp. To convert the timestamp

into a readable format, the `DsGetDateFromTimeStamp(ColumnName)` function is used. This function is used by the *ADS Docu User* report.

- DsPlainTextRTF: If you enter your text in a rich text box, the formatting is stored along with the actual text in the database. The DsPlainTextRTF function allows you to output the plain text in a text box, without considering the formatting.
  `DsPlainTextRTF(ColumnName)`

Permissions

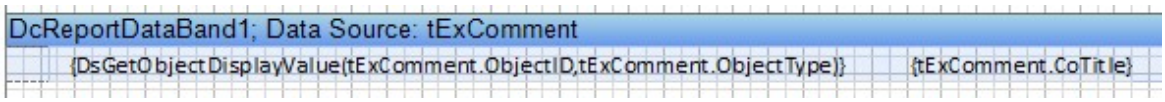
- DsAccessMask and DsACEFlag: These are special functions to determine the access mask or the ACE flags when scanning the permissions. These fields are located in the tACL table of the database.
- `{DsAccessMask(ColumnName)}`
- `{DsACEFlag(ColumnName)}`.



- GetBit: Using the *GetBit* function, you can individually evaluate the users' permissions to the directories. To show the permissions in a report, you can use the `GetBit(ColumnName,Bit)` function. For this purpose, the *Permission*column is used as the column name. For the Bit argument, enter the number that corresponds to the required permission. If the user has this permission, an "x" will be returned; otherwise, a space. This function is used in the *User (Resource), Directory (Resource)* and *Permission Analysis - Current View* reports.
  - Permissions and their Numeric Equivalents
  - 0 = Read
  - 1 = Write
  - 2 = Create directories
  - 3 = Read extended attributes
  - 4 = Change extended attributes
  - 5 = Execute
  - 6 = Delete subdirectories
  - 7 = Read attributes
  - 8 = Change attributes
  - 16 = Delete
  - 17 = Read permissions
  - 18 = Change permissions
  - 19 = Owner
- Share Permissions: For the output of share permissions, three functions are used. These are `IsFullAccess(ColumnName)` for full access, `IsChangeAccess(ColumnName)` for change access and `IsReadAccess(ColumnName)` for read-only access. Use the *Permission* column from the *tSharePermission* table as the column name. If the user has the respective share permission, an "x" will be returned; otherwise, a space.

Extensions

- DsGetObjectDisplayValue: Outputs an object linked with an extension. The values are stored in the tExtensions table and in the table for the respective extension, i.e. *tExComment, tExContract, tExFinance, tExPassword* and *tExReminder*. The syntax of this function is: `{DsGetObjectDisplayValue(ObjectId, ObjectType)}`



Exchange Server

- GetPublicFolderBit: This function is used for the output of the permissions to the *public folders* of the Exchange Server. To define the output of permissions, use the following syntax: `GetPublicFolderBit(ColumnName,Bit)`. Use the *ExchangePublicFolderPermission* column from the*tExchangePublicFolderPermission* table as the column name. For the Bit argument, enter the number that corresponds to the required permission. If the user has this permission, an "x" will be returned; otherwise, a space. The function is used by the *Public Folder Permissions*report.
  - Permissions and their Numeric Equivalents
  - 0 = Folder visible
  - 1 = Owner
  - 2 = Read elements
  - 3 = Create elements
  - 4 = Edit own elements
  - 5 = Delete own elements
  - 6 = Edit all elements
  - 7 = Delete all elements
  - 8 = Create subfolders
  - 9 = Folder owner
  - 10 = Publishing editor
  - 11 = Contributor
  - 12 = Author
  - 13 = None
  - 14 = Editor
  - 15 = Publishing author
  - 16 = Reviewer
  - 17 = Folder contact person
- GetExchangeMailboxBit: This function is used for output of the permissions to the *mailboxes* of the Exchange Server. To define the output of permissions, use the following syntax: `GetExchangeMailboxBit(ColumnName,Bit)`. Use the *ExchangeMailboxPermission* column from the*tExchangeMailboxPermission* table as the column name. For the Bit argument, enter the number that corresponds to

the required permission. If the user has this permission, an "x" will be returned; otherwise, a space. The function is used by the *Mailbox Permissions*report.

- Permissions and their Numeric Equivalents
- 0 = Full access
- 1 = Delete objects
- 2 = Read permissions
- 3 = Change permissions
- 4 = Change owner
- 5 = Send as
- 6 = External account

Licenses

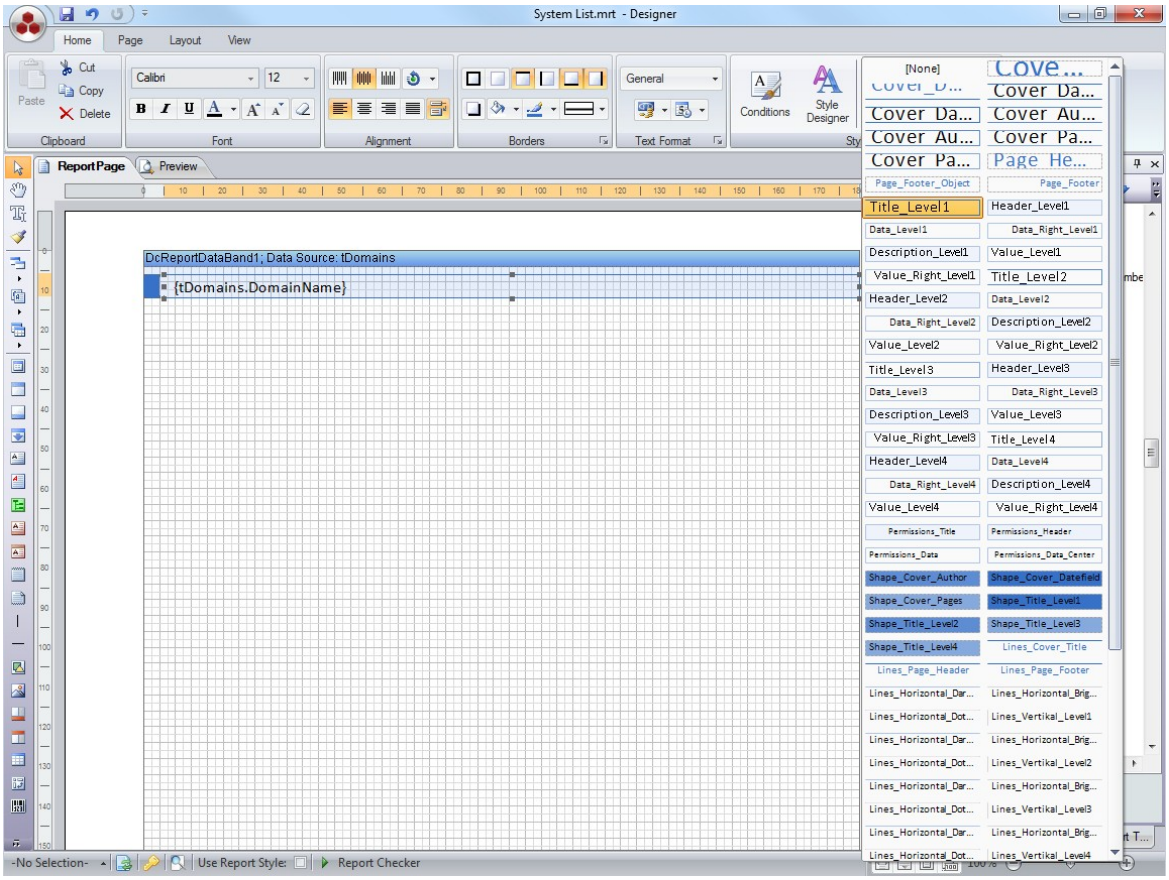
- Docusnap uses functions for license calculations. The value used is the SoftwareID from the tSoftwareProducts table.
  - Calculating the correction value
    - `{DsLicCorrectionValue(SoftwareID)}`
  - Counting the licenses purchased
    - `{DsLicAvailableLicenses(SoftwareID)}`
  - Counting the licenses in use
    - `{DsLicFoundCount(SoftwareID)}`
  - Calculating the upgrade and downgrade paths
    - `{DsLicPathCorrectionValue(SoftwareID)}`

2.7.3 Format Settings

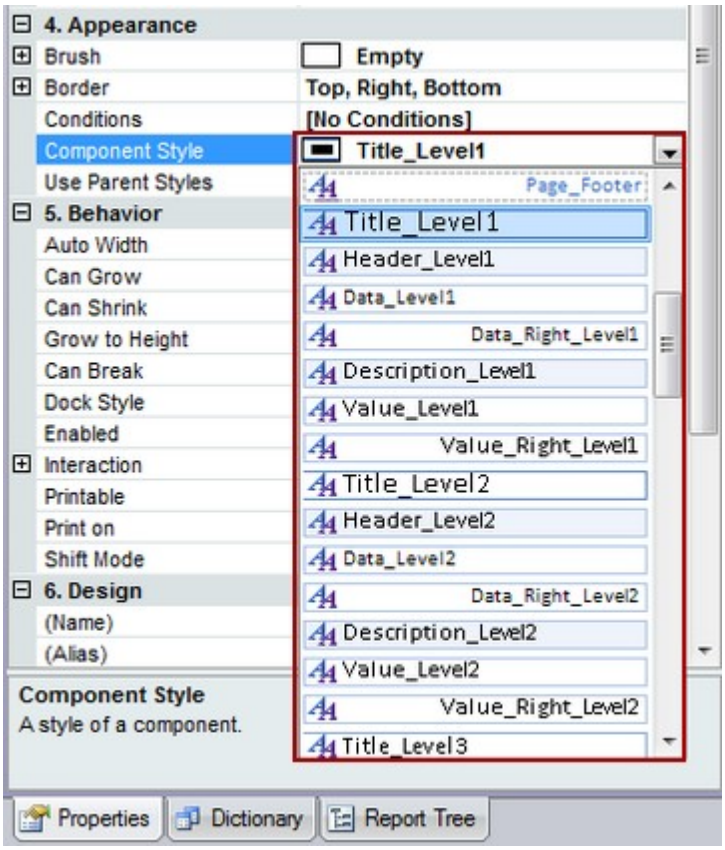
Formatting Reports in the Designer

When you open the Report Designer, Docusnap loads the current styles that have been selected in the *Designs and Styles* dialog. If you select a company, for which a different design was defined, from the status bar, that design will be loaded.

The styles are listed in the Style group of the Home ribbon.



To assign a style to a component, select the component first. Then, either select the associated style from the dropdown list of the ribbon or via the Component Style property on the Properties tab.

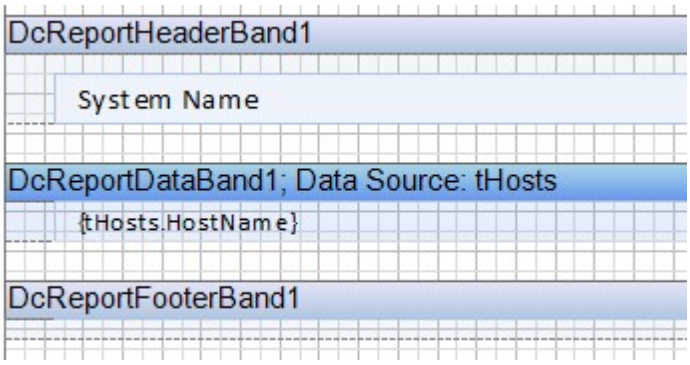


Styles

By default, styles have been defined for four different levels.

- Title: The Title_Level1 style is available for the title text and the Shape_Title_Level1 style for the shape title.
- Description: For the description output, the Description_Level1 style (left side) and a value using the Value_Level1 style (right side) are used. To align a value to the right margin, select the Value_Right_Level1 style.
- Headings: To format headings, you can used the Header_Level1 style.
- Data: Data can be formatted using the Data_Level1 and Data_Right_Level1 styles.
- Lines: Different styles are available for lines: Lines_Horizontal_Dark_Level1 and Lines_Horizontal_Bright_Level1. By using the Lines_Horizontal_Dotted_Level1 style, you draw a dotted line.

The other available styles can be used to format Permission Analysis reports and cover pages as well as headers and footers.



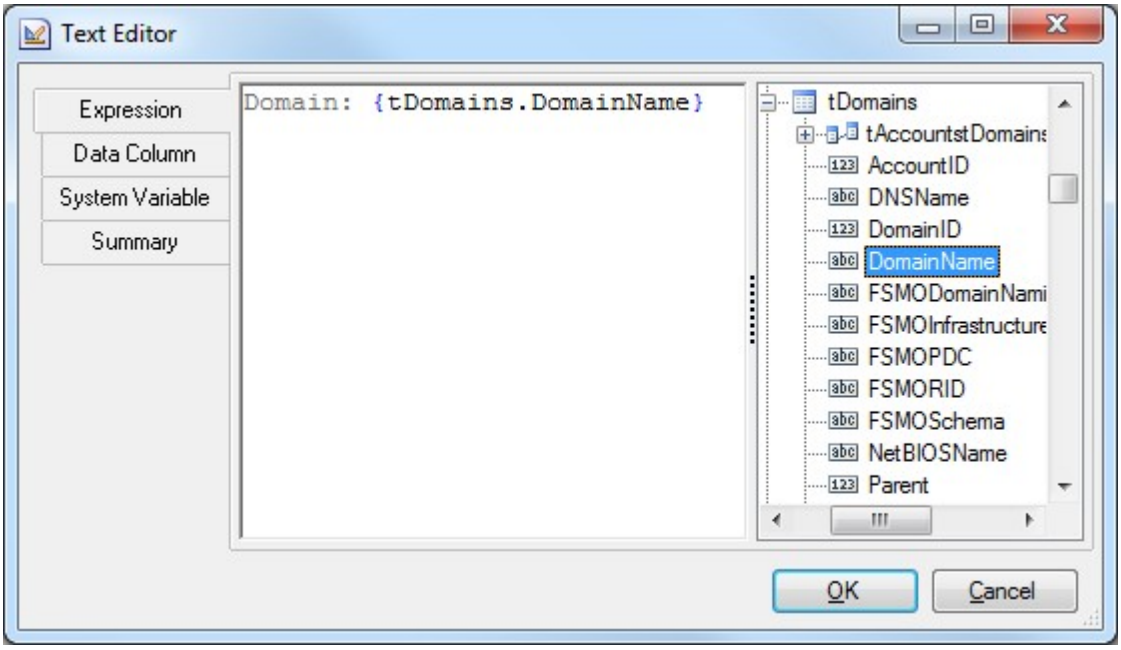
2.7.4 Advanced Structures

Hierarchies

In Docusnap, you can represent parent-child relationships between the tables. The dependencies between the tables can be implemented using relations or parameters. The parameters are explained in the Special Reporting Techniques section.

In the example below, we assume that an additional data band is required to which the tDomains data source will be assigned. First, add a Docusnap shape. For this purpose, use the *Shape_Title_Level1* style.

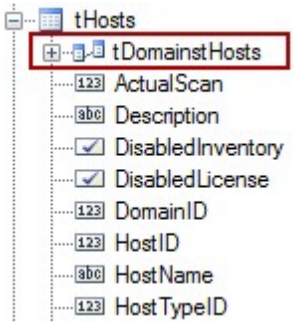
Then, create a text box for the domain name output. For the title, always use a single text box. In the Text Editor, enter "Domain" and add the *DomainName* column. The style to be used is *Title_Level1*.



Hierarchies: Relations

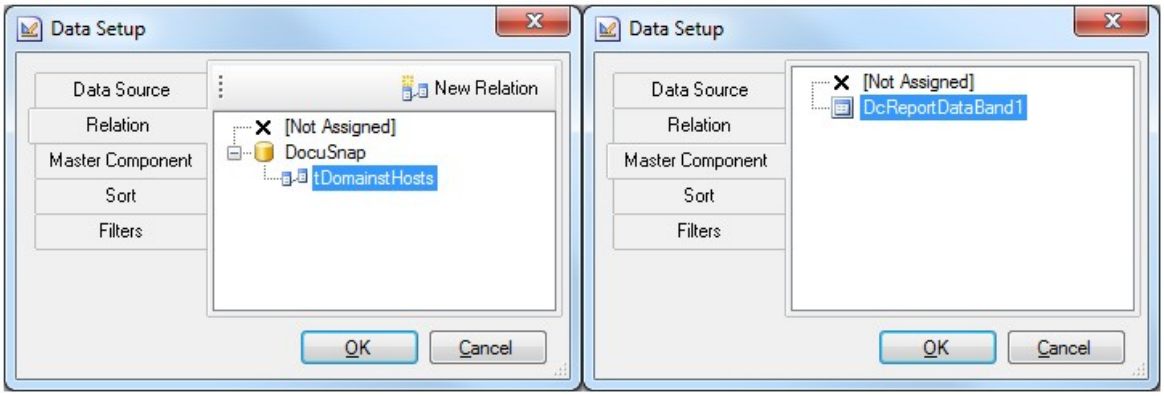
When you create a report, you will define all required relations. For a child data band, you need to select the relation and the parent data band as the master component.

In this report, the computers will be assigned to the domains. In the generated report, the first domain is shown in the first place, followed by all associated computers. Then, the next domain is shown with its associated computers, and so forth. In order to obtain this structure, a relation must be defined between the two data bands. Relations that exist in the database are also loaded into the Report Designer. Relations that do not exist can be created now. Relations always consist of a master component and a detail component. The detail component is the band that contains the data that will be associated. In this example, the data band with the computers will be used as the detail component. The master component is the data band that contains the domains. Relations will always be associated to the detail component. For this reason, enter the relation for the *tHosts* table. The icon indicates a relation. The name of the relation is composed of the name of the table for the master component and the name of the table for the detail component. In this report, the relation is named *tDomainstHosts*.



When you double-click the computers data band, the *Data Setup* dialog opens. There, you can select the desired relation. Click the *Relation* button to display all relations defined for this table. In this case, only one relation is listed: the relation with the *tDomains* table. Select the *tDomainstHosts* relation. Double-clicking a relation will select it and close the dialog. If you only click the relation once to select it, the dialog remains open. To make additional settings, select the relation using a single click.

Next, select the master component. The master component indicates the band on which this relation depends. The relation only represents the connection between the tables. If a report includes multiple bands, make sure to select the appropriate name. The name of the data band is displayed on the left in its blue title bar. In this report, there is only one other data band. Select this data band. Click the *OK* button to apply the relation and the master component to the detail component.

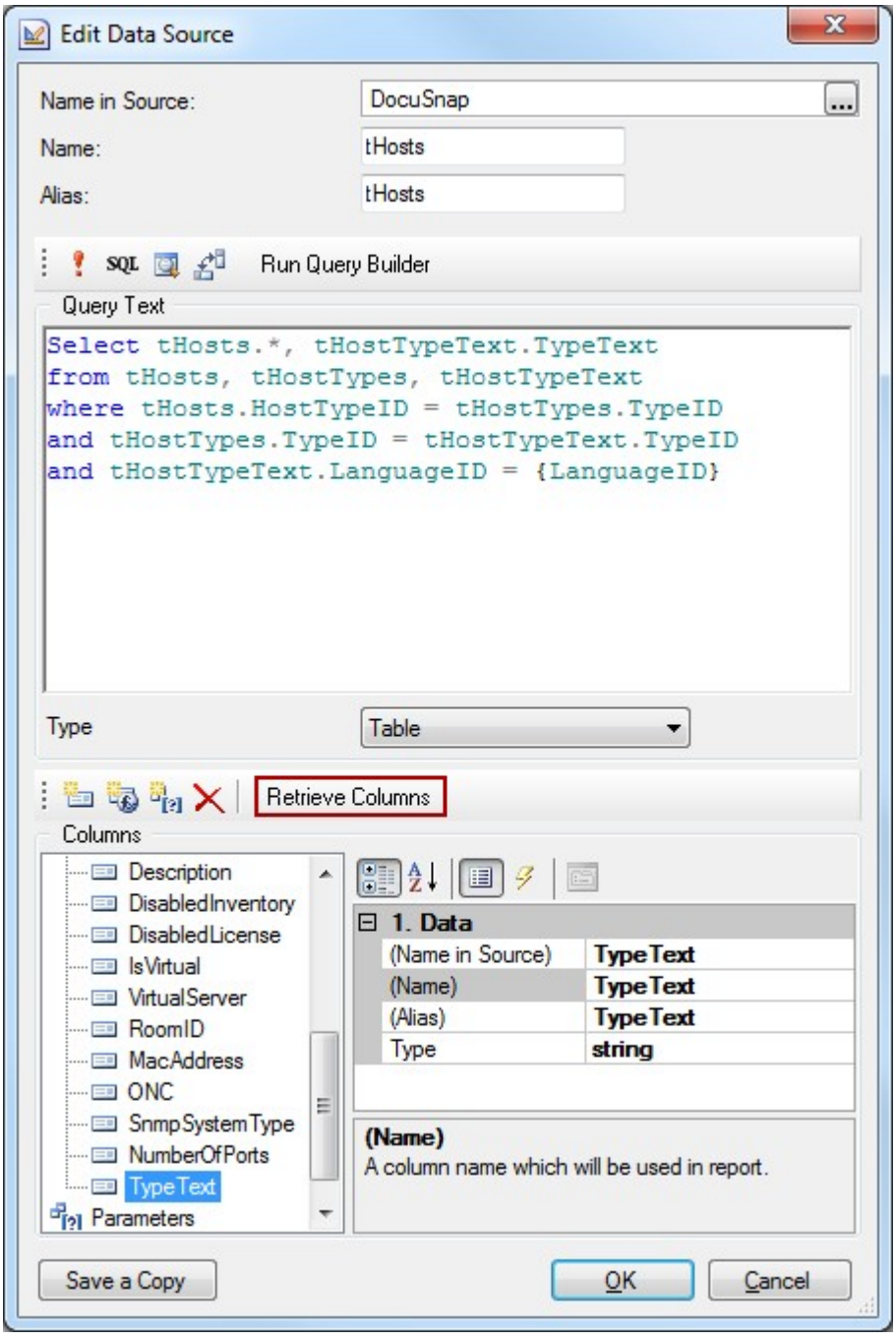


Showing Types in Multiple Languages

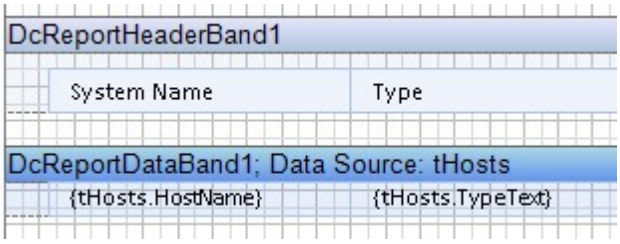
- Since Docusnap uses two languages, all types are stored in English and German. In the database structure, two additional tables are used for this purpose besides the table that contains the actual data: The Type table contains the table name and the TypeText table contains its name in English and German.
- For each report, Docusnap creates the LanguageID variable. This variable contains either 0 for German or 1 for English, depending on the language that has been selected in that report. By means of this variable, the types can be shown in the desired language.
- In order to show the type text, the SQL statement for the data source needs to be modified. Example of the new SQL statement based on the Host type:

```
Select tHosts.*, TypeText as Type
from tHosts, tHostTypes, tHostTypeText
where tHosts.HostTypeID = tHostTypes.TypeID and tHostTypes.TypeID =
tHostTypeText.TypeID and
LanguageID = {LanguageID}
```

- To add the additional column, click the *Retrieve Columns* button.

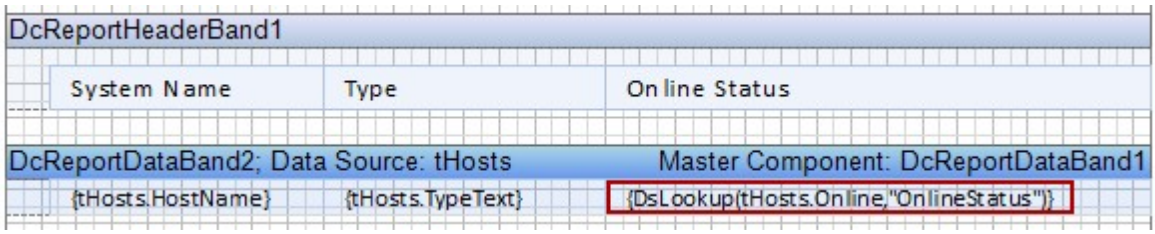


- Then, Docusnap copies the text boxes for the text and the heading. Finally, you can edit the heading and the selected column, if required.



2.7.5 Reference Values

Many tables contain only numbers. To express these numbers as words, the *tSysInitials* table is available in the Docusnap database. The *DsLookup* function can be used to express the values from these tables as literals. Enter this function into a text box located in a data band. The function must be surrounded by braces "{}", otherwise the Designer will not recognize it as a function. To show the online status of the *tHosts* table in the report, add a text box and type the following function into this field: `{DsLookup(tHosts.Online,"OnlineStatus")}`. For an overview of the reference values, refer to the Reference Values section in the appendix.



2.8 Special Reporting Techniques

This section will explain the use of parameters. The advantage of parameters over relations is that the reports can be executed more quickly. The parameter is added to the SQL statement. This results in only the filtered data being retrieved from the database.

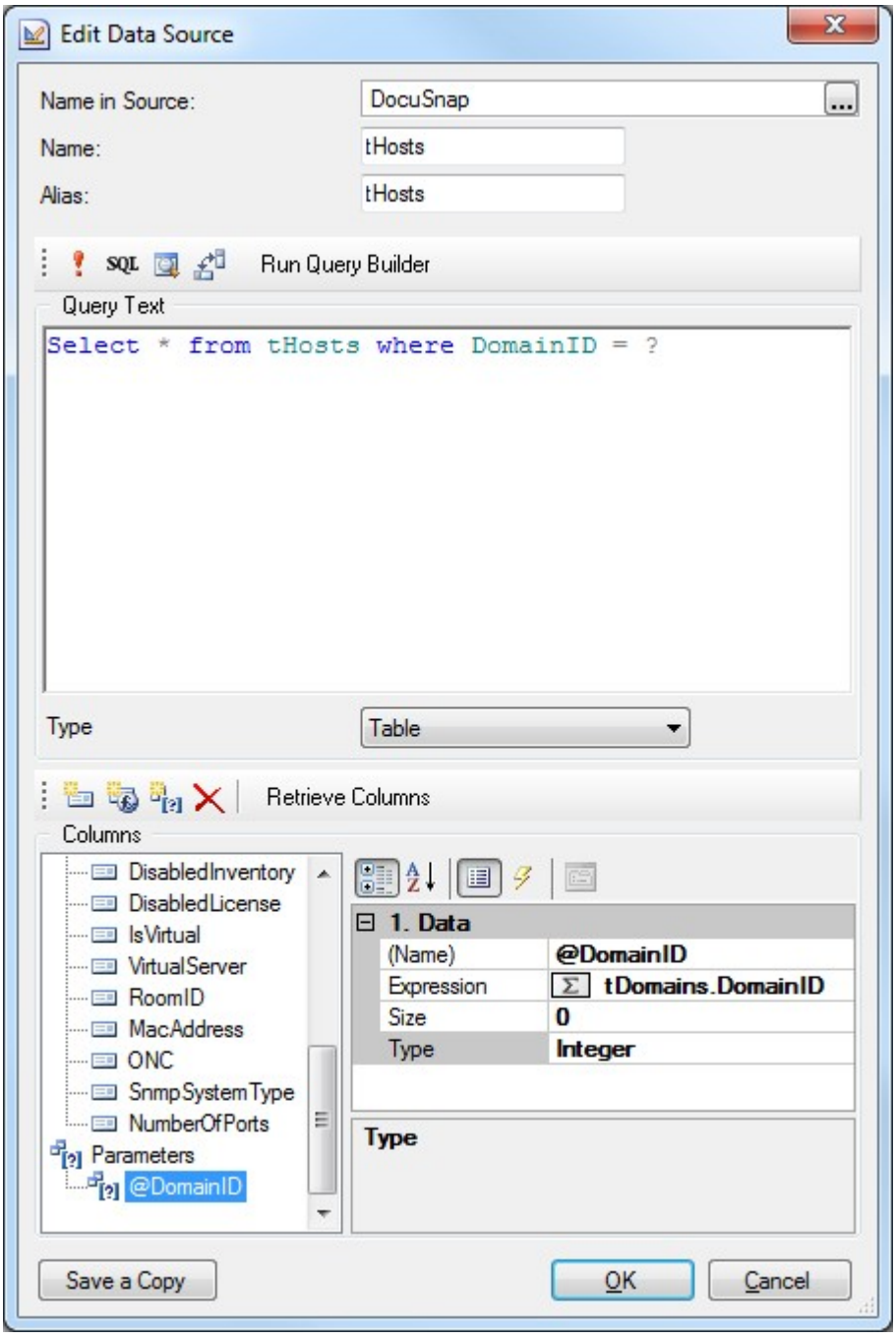
In addition, grouped reports and hierarchical reports will be covered in the following sections. In the Sub-Reports section, you will learn how to use a sub-report to generate a single row that shows data from multiple tables.

2.8.1 Parameters

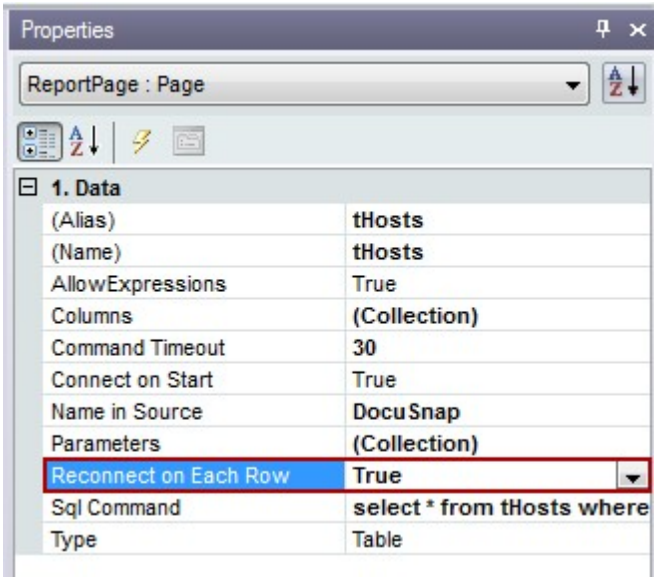
Hierarchies: Parameters

If you use parameters instead of relations to represent the dependencies between tables, reports can be executed more quickly. You can add the required parameter to the SQL statement. The result is that only the filtered data is transferred from the database to the client. The parameter is represented in the statement by a question mark (*?*) placeholder.

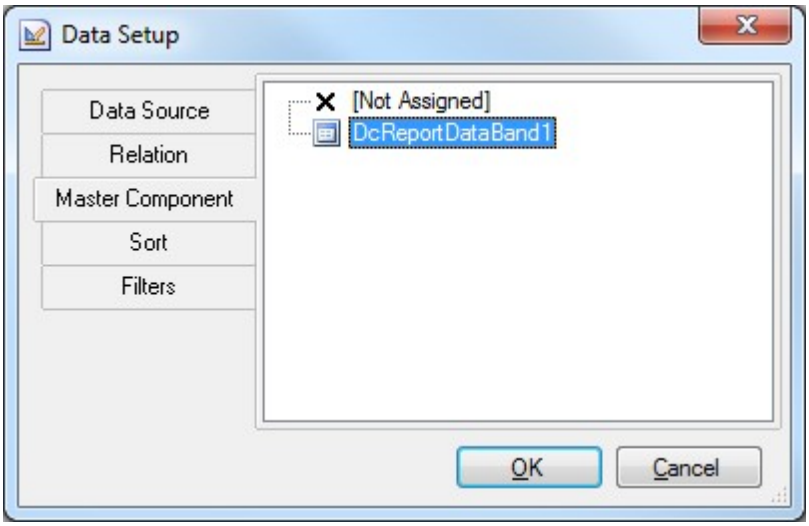
After entering the statement, create the parameter and select the expression from the parent table. If you specify multiple parameters, they will be used in the order in which they are listed.



For the table, the *Reconnect on Each Row* property must be set to *True* on the Properties tab.



When defining the data band, the parent data band must be defined as the master component. In the parent data band, select the parent table from which the parameter was selected, as the data source.



2.8.2 Grouped Reports

Computers grouped by Operating System

This sample report will include all computers grouped by their operating system.

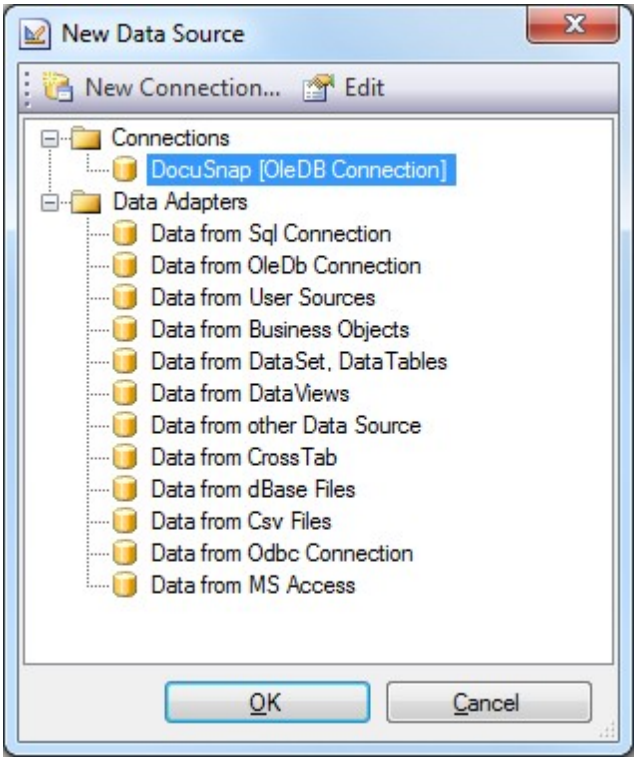
The computers are found in the *tHosts* table and the operating systems in the *tDocuWindows* table. Due to the relation created between the *tHosts* and *tDocuWindows* tables, a given computer may have multiple entries in the *tDocuWindows* table, because it may have been scanned several times. In a report designed on the basis of relations, it is possible that multiple detail components exist for each master component. In this report, the *tDocuWindows* table would be the master component and the *tHosts* table would be the detail component, as the computers will be assigned to the operating systems. Since there may be several

entries in the *tDocuWindows* table (master component) for each computer (detail component), this report cannot be resolved using native database relations. In this case, you need to create a virtual table using an SQL statement. It will invert the physical structure and generate a corresponding list.

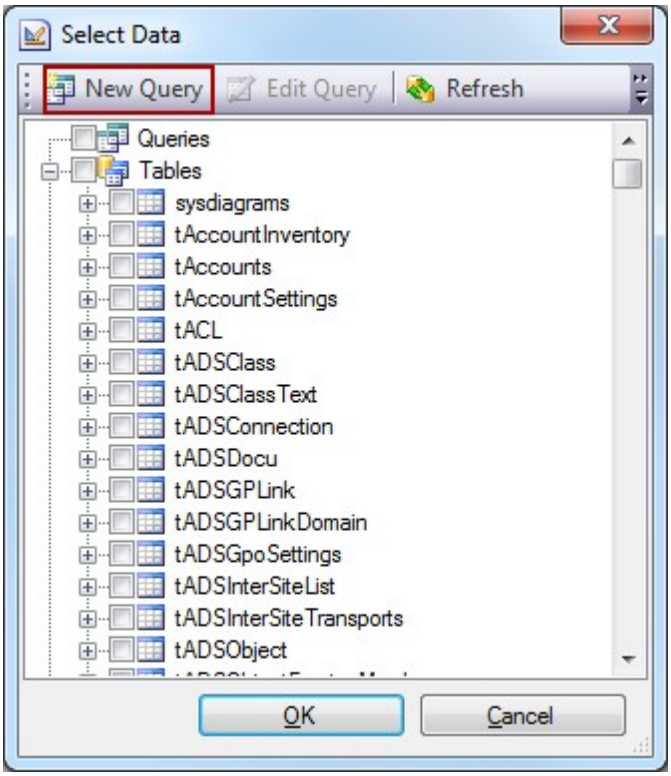
SQL Statement

When you create a new report, all existing database tables are loaded into the report. For this report, however, a table is required that does not exist in the meta tables. For this reason, you must create this table in the Report Designer using an SQL statement.

First, add a new data source. To do so, go to the Dictionary tab and click the  button or right-click to open the context menu and select New Data Source. As the connection, select DocuSnap [OleDB Connection]. Click *OK* to confirm the connection. A dialog opens where you can select the tables to be connected.



As an alternative, you can open this dialog by double-clicking the connection. From this dialog, you can import newly added tables to the Dictionary that were not yet present in the database when the report was created. For this sample report, an additional table is required. Click the *New Query* button to open the *New Data Source* dialog.



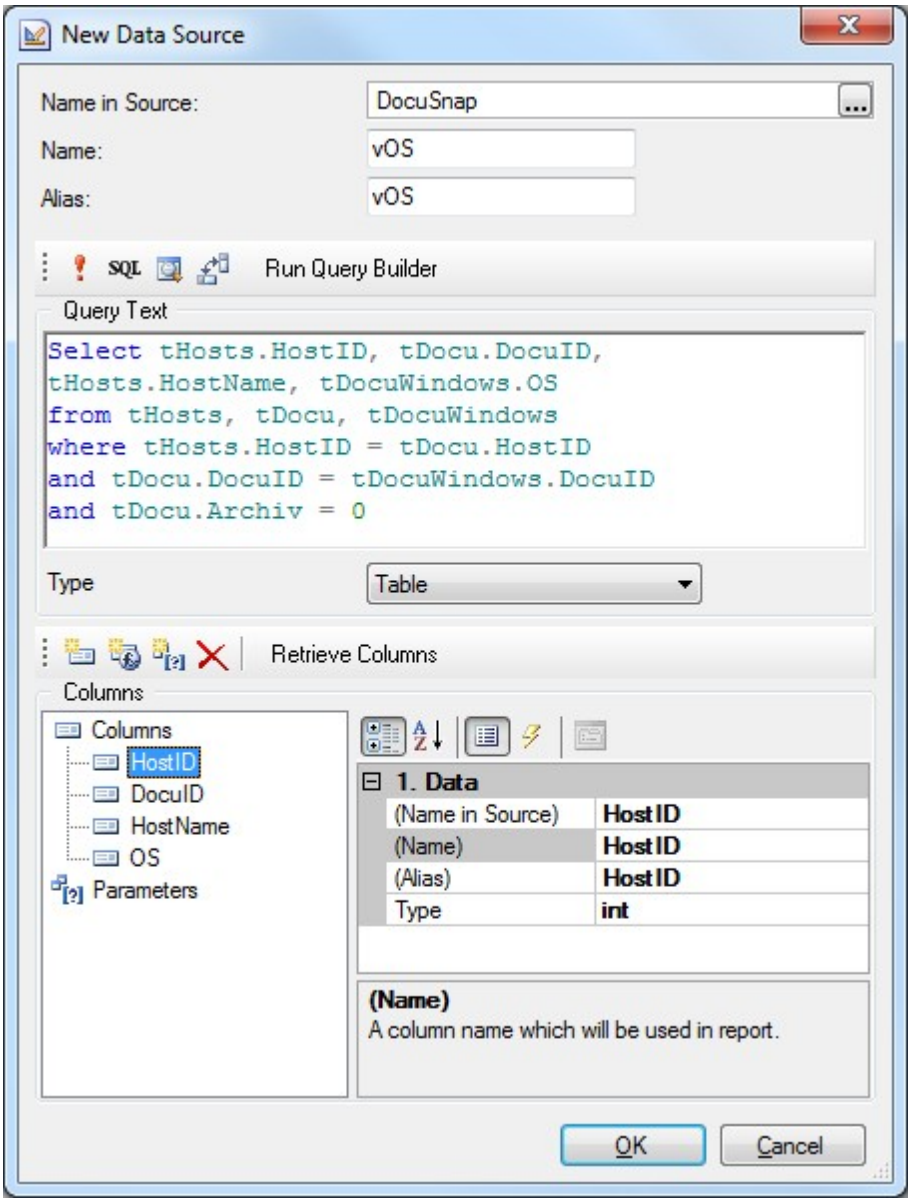
From this dialog, you can create the new data source. The correct database has already been entered as the name of the data source. The table name can be chosen freely. We recommend to prefix the name of the table with a lowercase "*v*" for "virtual", to distinguish this table from those already created. For this report, name the table *vOS*. The Alias field is automatically populated with the same name. By clicking the button, you can validate the SQL statement you entered. To open a text editor where you can enter the SQL statement, click the SQL button. The data retrieved by the SQL statement can be displayed by clicking the button. Using the button, you can change the dialog view. Enter the SQL statement in the *Query Text* field. You can enter any SQL statement desired in the *Query Text* field as long as it relates to tables and columns that exist in the database. When you are done entering the statement, add the columns by clicking the *Retrieve Columns* tab. The table columns are displayed in the *Columns* field. When you select one of the columns, its properties are displayed to the right of the selected column. When you add a column, the primary key is defined as an *int* data type. For existing database tables, the primary keys are of the *long* data type. When you create a relation, make sure that the related columns are of the same data type. For this reason, change the primary key for new tables to the *long* data type.

For this report, a table that combines data from the *tHosts* and the *tDocuWindows* tables is required. Currently, there is no direct relation between the *tHosts* and *tDocuWindows* tables. The connection will be made through the *tDocu* table. The *tDocu* table is linked with the *tHosts* and *tDocuWindows* tables. The *tHosts* table is linked with the *tDocu* table by means of the *HostID* column. The relation between the *tDocu* and *tDocuWindows* tables is created through the *DocuID* column.

For the new table, not all columns from these tables will be needed. Therefore, select only some of them. For the output, the computer name (*Hostname* column) and the operating system (*OS* column) are required. It is a good idea to specify the primary keys of the tables for the output fields of the SQL statement, although they do not actually appear in the report. In addition, the *tDocu* table has a special feature. Assume that you want to limit the report to the most recent data. However, each computer may have been scanned multiple times. To obtain only the current information, you can use the *Archiv* column in the *tDocu* table. For the most recent scan in the *Archiv* column, the value 0 (zero) will be set. By using the *Archiv = 0* Where condition, only the current data will be added to the new table. The resulting SQL statement connects all necessary tables:

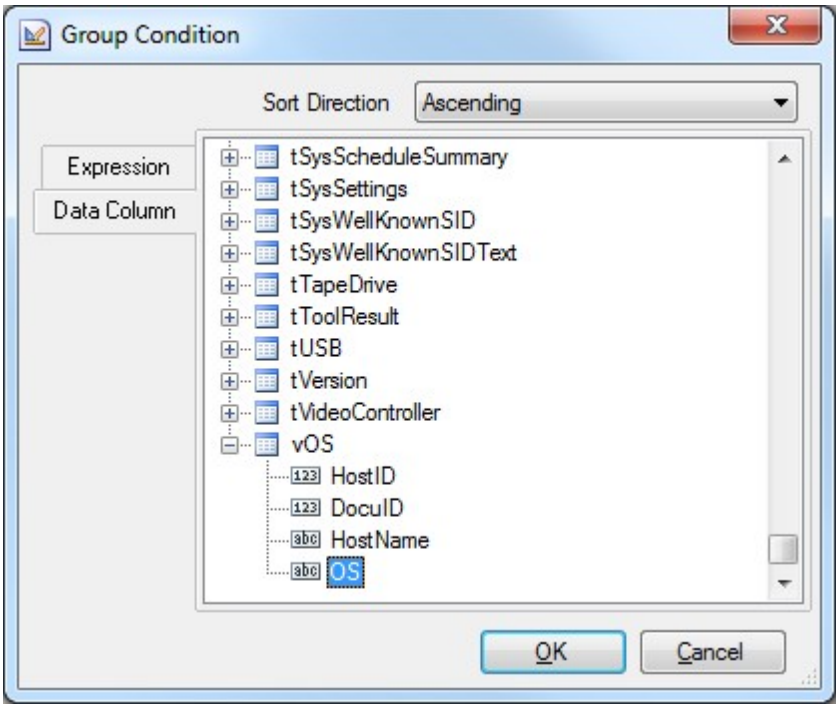
```
Select tHosts.HostID, tDocu.DocuID, tHosts.Hostname, tDocuWindows.OS
from tHosts, tDocu, tDocuWindows
where tHosts.HostID = tDocu.HostID and tDocu.DocuID = tDocuWindows.DocuID and
tDocu.Archiv = 0
```

When you click the *Retrieve Columns* button, the columns you indicated in the SQL statement will be retrieved from the database. Both *HostID* and *DocuID* are primary keys. For this reason, the data type needs to be changed to *long*. Click the *OK* button to close the dialog. Next, select the newly created table, *vOS*, under the *Queries* node in the *Select Data* dialog. Click *OK* to add the table to the Dictionary. The *vOS* table now appears on the Dictionary tab.

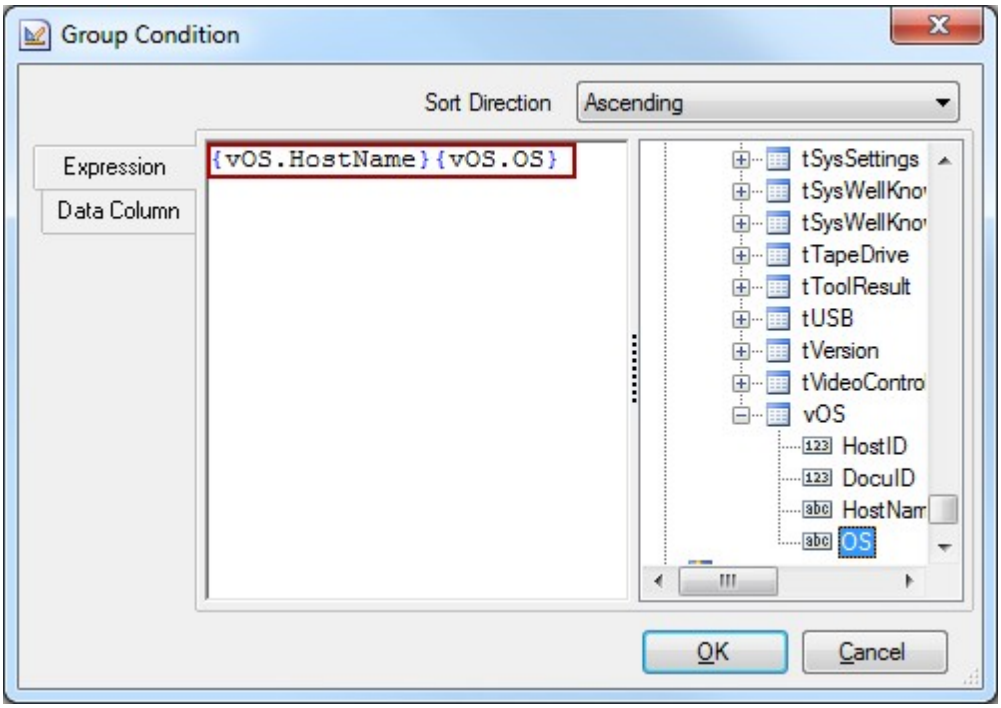


Creating a Group

To group the data of a table, add a group header band to your report. Group header bands can be added from the Toolbox. Open the the Group Condition dialog to do so. Here, you can specify the column on which to group your data. For this sample report, select the operating system (*OS* column) from the *vOS* table. You can also specify whether the group should be sorted in ascending or descending order or not at all.



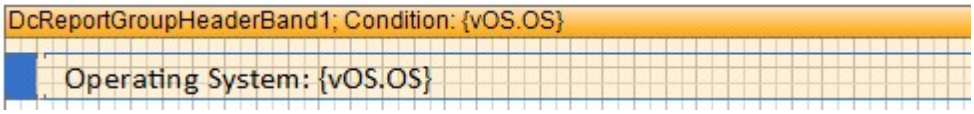
If you want to use more than one column for grouping, open the *Expression* page. On this page, you can add multiple columns. Thereby, only those records that match the selected expression will be output as a group. To apply the condition to the group header band, click *OK*.



Group Output

By selecting the condition, you can determine which criterion will be used for grouping. The group header band is of course still empty. The group header band will be used for the title of this report. For this reason, the format setting for the

title will be used.



This ensures that the grouping criterion will be shown at the top.

Child Band

The child band is an extension of the band preceding it. In this report, a child band is used as an extension of the group header band. Following the title, a header band is often added to provide the heading for the data band. Please note that a header band for the headings cannot be inserted after a group header band, because the bands would not be shown in the correct sequence: The headers would not be placed after the title and before the data. For this reason, a child band is used for the headings of the data band. The child band is always shown after its parent band. In this report, the band with the headings will be shown after each group header band.

Create a text box, enter the "Computer" heading into it and assign the *Header_Level1* style.

Data Band

In order to output the data for the group header band, a data band must exist for which the same data source has been specified as in the condition for the group header band. In order to be grouped by the group header band, this data band does not need a defined relation or a master component. As the data source, select the same data source that was used for the group header band. Select the *vOS* table and the HostName column for this report.

In the next step, create the text box that will show the database content. When you add this field, the Text Editor opens. In the Text Editor, you can select the data fields either on the *Expression* or the *Data Column* page. To add a field from the *Expression* page, double-click it. On the *Data Column* page, click the desired field. For this report, select the HostName column from the vOS table. The style to be used is *Data_Level1*.

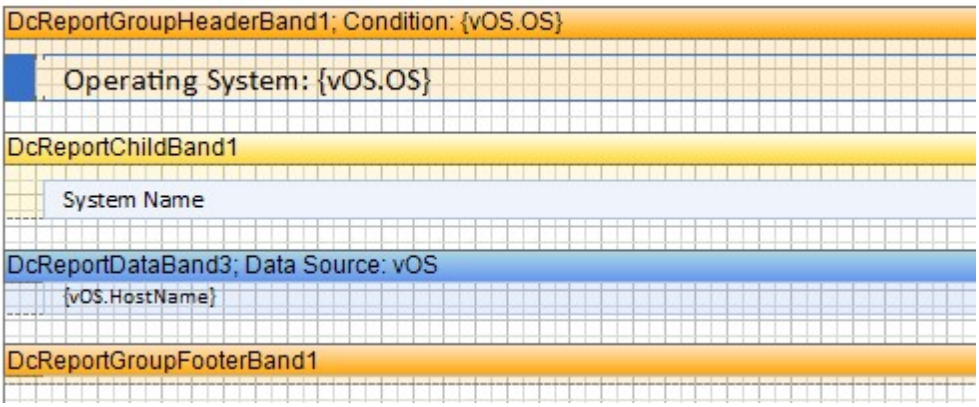
Group Footer Band

Finally, a group footer band will be added. The group footer band is the closing item of the group.

Draw a closing line for the data output on the group footer band. Assign the *Lines_Horizontal_Dark_Level1* style to this line.

Finished Report

The header band contains the text box that indicates the operating system. The name and the condition appear in the left corner of the group header band. The "Computer" heading will be shown on the child band. The data band includes the text box with the *HostName* column. The data band name and the data source appear in the upper left corner of the data band. The closing line is shown on the group footer band.



2.8.3 Hierarchical Reports

Listing the Directory Structure

The directories of all computers in the domain are stored in the *tDirectories* table. A folder that does not have a parent directory will be identified by the value -1. For each other folder, the *DirectoryID* of its parent folder is listed in the *Parent* column. This enables you to build a hierarchy. This hierarchy will be created in the Report Designer using a hierarchical data band.

Hierarchical Data Band

When you add a hierarchical data band, the *Data Setup* dialog opens. For this report, select the *tDirectories* table. Then, specify the criteria to build the hierarchy. On the Properties tab, set the *Key Data Column* property to *DirectoryID* and the *Master Key Data Column* property to *Parent*. The parent value identifies the "parent" of the top folder. If you do not enter a value for this property, this column, identifying the parent record, must be empty for the top level directory in the hierarchy.

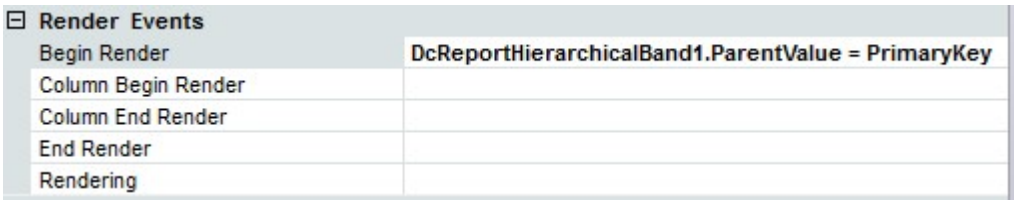
For the *Indent* property, specify the number of millimeters for indenting the next level. If only the first text box should be indented, the *Locked* property must be set to *True* for the remaining fields. Using the Headers and Footers properties, you can select the header band and the footer band to be shown before each hierarchy level.

Parent Value

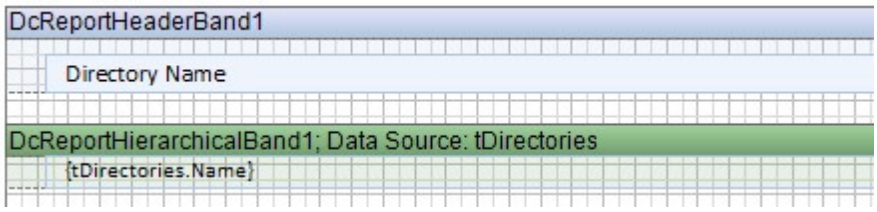
In addition to a number, the primary key or a column from another table may be

specified as the parent value. However, this value cannot be entered in the *Parent Value* property on the Properties tab, but rather must be defined as an event. Select the page by clicking the white margin or an area of the page that does not have a band.

Switch to the Events properties by clicking the ⚡ button on the Properties tab. The parent value can be defined using the *Begin Render* property. When assigning, first specify the name of the hierarchical band and then use *.ParentValue* to get the parent value. Finally, assign the desired value using the equals sign (=). In this report, it is called DcReportHierarchicalBand1.



Finally, you can add text boxes for user input.



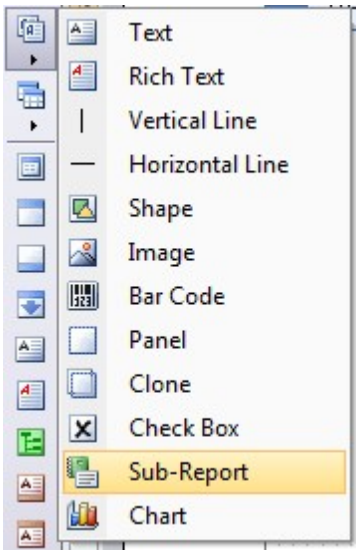
2.8.4 Sub-reports

To show the contents of a table and its child table in a single row, you can use a sub-report.

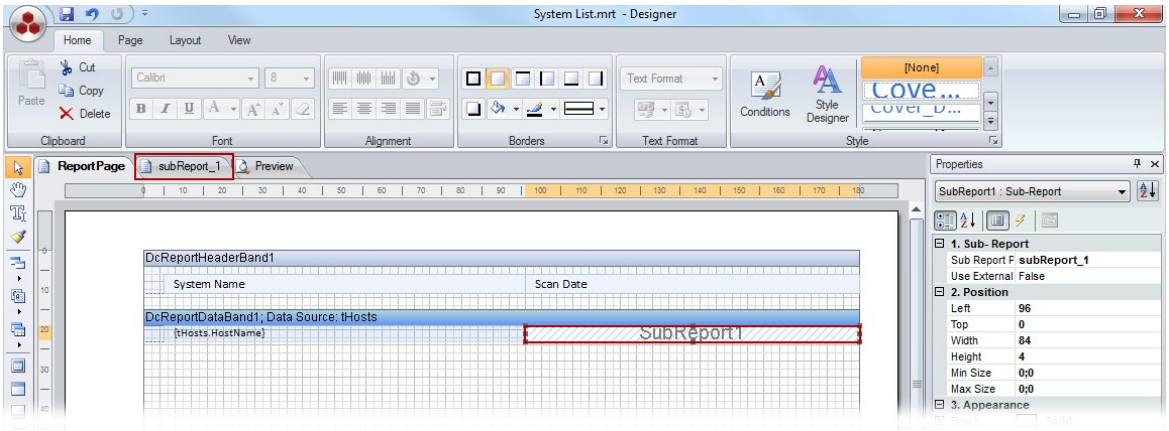
In this example, the dates of all inventory scans performed on this computer are shown. The data is found in the tHosts and tDocu tables.

First, create a header band with the *System Name* and *Scan Date* headings. Then, add a data band using the *tHosts* table as the data source and add a text box containing the *HostName*.

Next, create a *sub-report*. To create the sub-report, use the icon from the Toolbox.

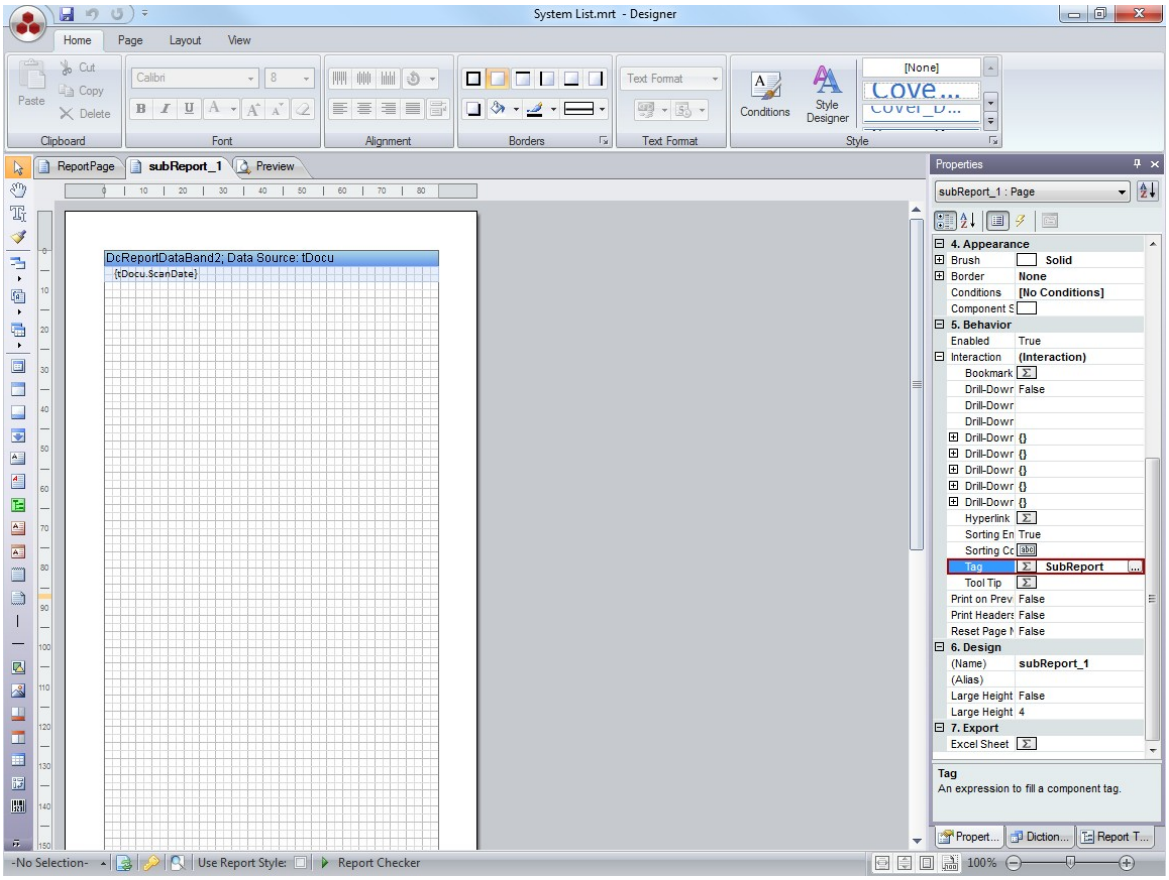


Docusnap automatically adds an additional page where you can define the sub-report.



Create the child data band, in this example the data band with the *tDocu* data source, on the sub-report page. Add the text boxes for the data of this child data band here. In this example, add a text box for the *Scan Date* from the *tDocu* table.

When generating a report, Docusnap creates all defined pages with a cover page, a footer and a header. However, since the sub-report only involves a "page" that will be output as part of another table, neither a cover page nor a header or footer should be shown for it. For this reason, enter the word *SubReport* in the *Tag* property of the *Interaction* group on the Properties tab of the sub-report. This ensures that these "pages" will neither be given a cover page nor a header or a footer.



As the master component of the data band, you can use the data band where the sub-report will be inserted. Create the connection to the master component using a relation or a parameter. After the *tHosts* and *tDocu* tables have been connected to the database, the report can be executed.

itelio

Scan Date

System Name	Scan Date
SMPS0001	08.12.2011 09:51:46
	23.01.2012 16:33:25
	08.02.2012 14:07:38
	16.02.2012 09:34:00
SMDC0001	06.07.2011 16:06:33
	18.08.2011 13:24:05
	21.11.2011 15:14:16
	08.12.2011 09:51:46
NKNG0002	12.03.2011 09:33:19
SMEX0005	21.11.2011 15:14:16
	08.12.2011 09:51:46
	23.01.2012 16:33:25
	16.02.2012 09:34:00
WMWS0333	06.07.2011 16:06:33
	18.08.2011 13:24:05
	21.11.2011 15:14:16
	08.12.2011 09:51:46
WMWS0113	06.07.2011 16:06:33
	18.08.2011 13:24:05
	21.11.2011 15:14:16
	08.12.2011 09:51:46

2.8.5 SNMP Reports

Docusnap relies on MIBs when performing the inventory scan for SNMP devices. For the output of the SNMP device data in a report, tables are used that are populated with the retrieved data. These tables are for example used for the Summary *SNMP report* that lists the data for each SNMP device or for the *Active Network Components* report that indicates the SNMP devices of a domain. Adding SNMP tables only works for reports whose primary key corresponds to the *DocuID* or the *DomainID* of the respective SNMP devices.

To add the additional tables for SNMP devices, you need to create variables whose names start with *vSNMPMib*. First write the text *SNMP:* followed by the MIB (in parentheses), and, separated by a comma, the name, into these variables. To separate the MIBs, enter a semi-colon. If you add the *ID:DomainID* after the closing parenthesis, the tables will be output for all SNMP devices existing in the domain.

This results in the following syntax:

```
SNMP:(1.3.6.1.2.1.4.20.1.1,IPAddress;1.3.6.1.2.1.4.20.1.3,
SubnetMask;1.3.6.1.2.1.4.20.1.2,InterfaceID)
```

```
SNMP:(1.3.6.1.2.1.4.20.1.1,IPAddress;1.3.6.1.2.1.4.20.1.3,
SubnetMask;1.3.6.1.2.1.4.20.1.2,InterfaceID)ID:DomainID
```

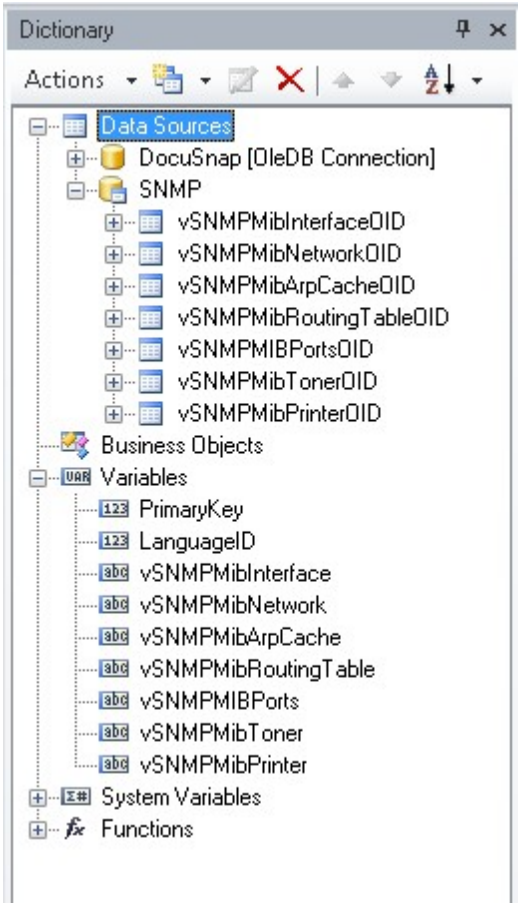
To ensure the output of values that exist only once for each SNMP device, precede the statement with *SNMP-Single*.

Use the following syntax:

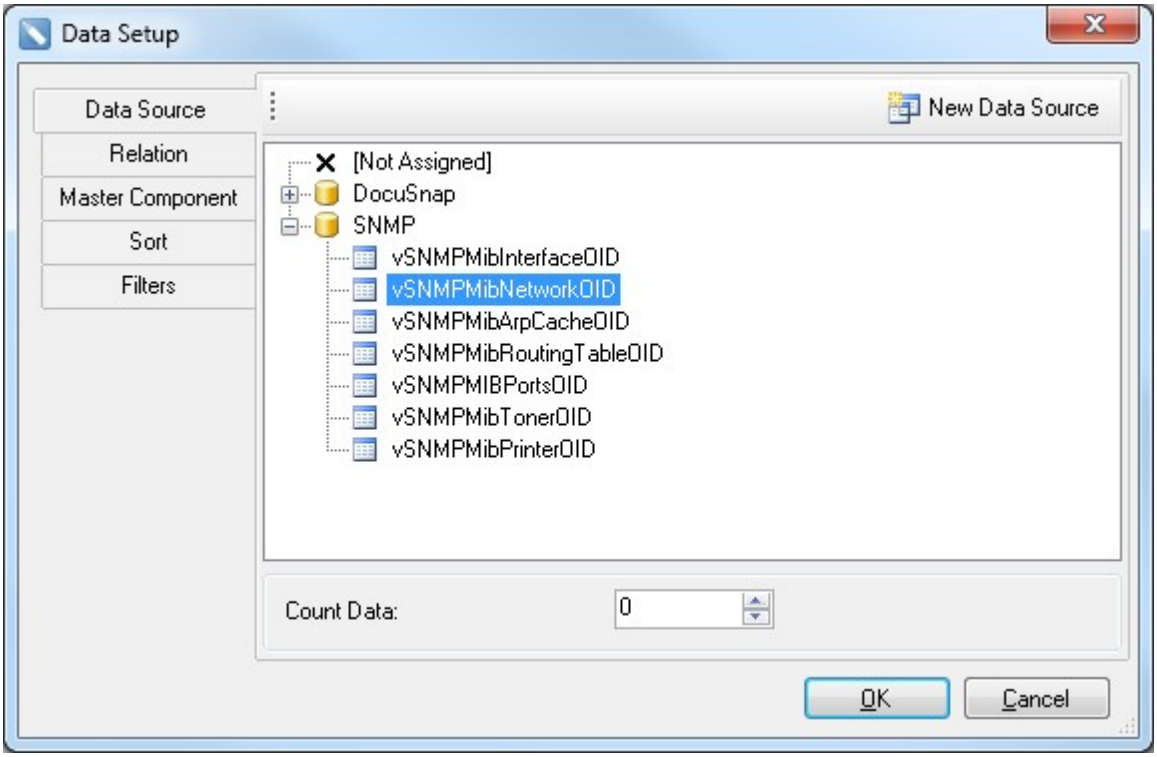
```
SNMP-Single: (1.3.6.1.2.1.43.8.2.1.14.1.1,Manufacturer;1.3.6.1.2.1.43.5.1.1.17.1,
SerialNumber)
```

```
SNMP-Single: (1.3.6.1.2.1.43.8.2.1.14.1.1,Manufacturer;1.3.6.1.2.1.43.5.1.1.17.1,
SerialNumber)ID:DomainID
```

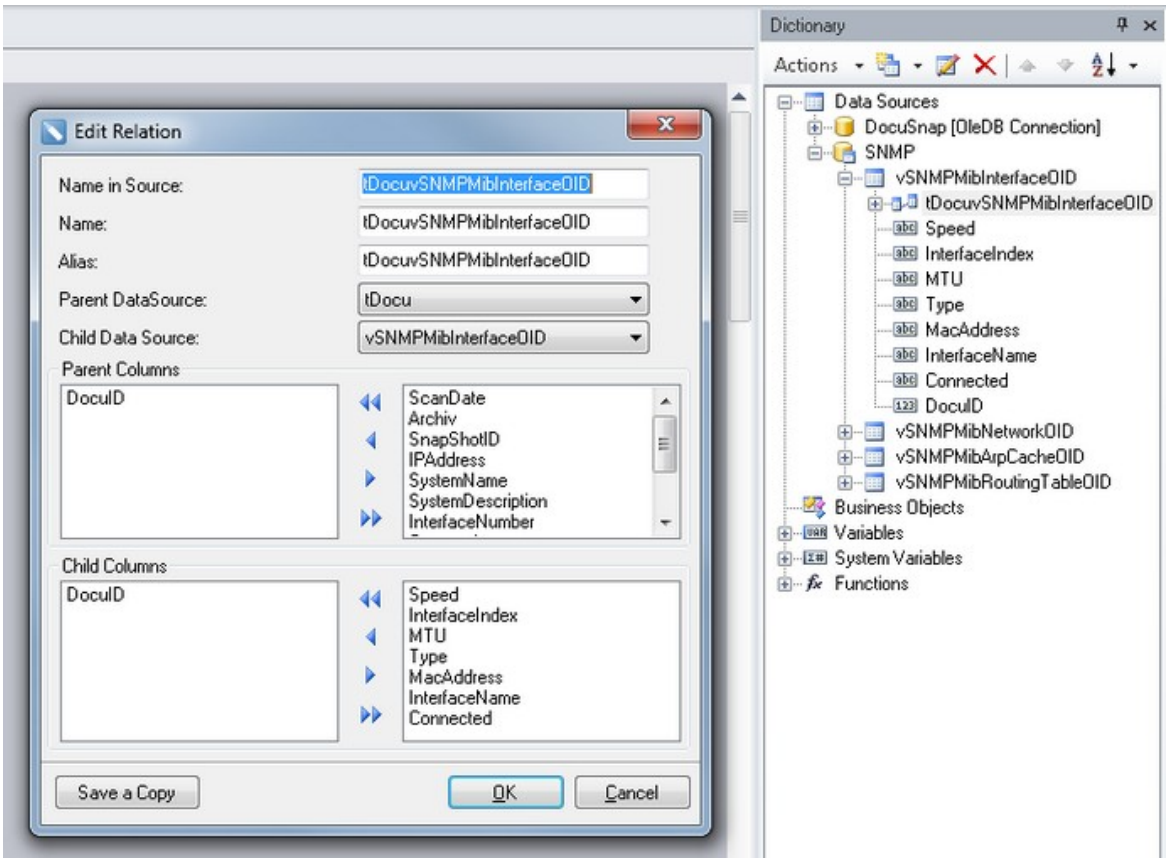
Once you have created all required variables, close the report and then open it again. This ensures that the Report Designer will create the tables for SNMP when loading the report. In the Report Designer, the tables are listed under the *SNMP* data connection node that is found below the Data Sources node.



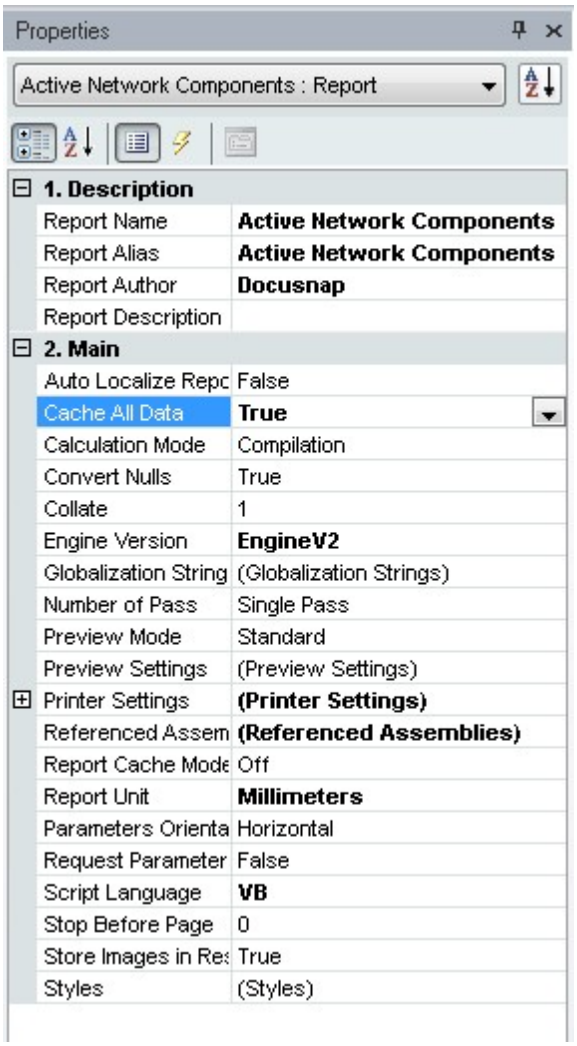
When creating the report, you can select the SNMP tables as data sources for data bands.



Using relations, you can create a hierarchical structure. Every SNMP table has a *DocuID* field which can be used to define a Relation to other tables.



Since the relation spans two different data sources, you must set the *Cache All Data* property in the report properties to *True*.



2.9 Dialog

It is possible to filter the data output in the report by adding a preliminary dialog.

You can define a dialog where the user can e.g. select the system for which the report should be created or specify the period that should be considered for the report.

To use a dialog in your report, you first need to create a new form in the Report Designer by right-clicking the horizontal area to the right of the Preview tab and selecting *New Form* from the context menu.

Now, the toolbox displays the components that can be added to the form. The *Properties* window displays the settings for the selected component.

2.9.1 Components

For each of the components, you can specify a name in the *(Name)* field of the Properties window. This name is required to access the value of this component later.

Label

The label is used to describe another control. For example, you can specify which values have to be entered into a text box.

Text Box

Users can enter any desired text in a text box. To later reference the text entered into the text box, use *(Name).Text*.

Group

You can use group controls to organize other components in a clearly arranged form layout. From a functional point of view, there is no difference between grouped components and components that are arranged directly on the main screen. Use the *Text* property to specify the text to be displayed as the group title.

Button

For each form, you must add at least one button whose *Dialog Result* property has a value other than *None* or *Cancel*. The *OK* or *Yes* dialog results are best suited to apply the values from the form to the report.

Checkbox

To enable the input of Boolean values, add checkboxes. If the checkmark is set, the value of the component is *True*, if the checkbox is empty, the value of the component is *False*. The *Checked* property defines whether the checkmark will be set by default or not. To later reference the status of the checkbox, use (Name) .checked.

Radio Button

Use radio buttons if you want to provide a set of options excluding each other (i.e. only one option may be selected at a time). If you want to add multiple radio button sets, either arrange them in group boxes or panels. For each set of radio buttons, one radio button should be defined with the *Checked* property set to *True*. In the report, you can use the *(Name).checked* expression to find out whether the button was selected.

Combo Box and LookUp Box

For entries whose allowed values can be selected from a list, so called combo boxes or lookup boxes are available. These components are suitable for all cases where a limited number of valid values is available which may change dynamically. As the source for the values of the combo box, you can either define a data column or enter static values.

In the lookup box, a key is stored in addition to the value. This enables you to filter not only on the selected name, but, e.g. also on the primary key of the selected value.

In the Properties window, select the data column, e.g., *tHosts.Hostname* under *(Data Bindings) - Items*. When adding a lookup box, additionally select the data column to be used as the key, e.g. *tHosts.HostID*. You can enter a static list of values into the *Items* and *Keys* properties which are not grouped under *(Data Bindings)*.

Then, you can specify the *(Name).selectedItem* and *(Name).selectedKey* expressions to use the selected values or keys in the report.

Checked List Box

A *checked list box* lists values which the user can select by enabling the checkbox to the left of the corresponding value. In the Properties window, under *(Data Bindings) - Items,* select the data column whose values will be made available for selection. Using the *Check on Click* option in the *Behavior* category of the Properties window, you can define if a value is enabled by simply clicking on it or if the associated checkbox needs to be checked to enable the value. In the report, you can reference the selected values by specifying *(Name).selectedItems*.

Date Time Picker

Date-time pickers are used to select a date. Above all, this component is required if the user wants to restrict the report output to a certain period of time. In the report, specify *(Name).value* to reference the selected date. When you select the date, the current time is added automatically. If you only want to use the date, specify *(Name).value.date*.

Numbers (Numeric Up Down)

Use the *Numeric Up Down* component to make sure that only numbers will be entered. In addition, you can define a minimum and a maximum value. You can enter a default value using the *Value* property. To reference the number entered by the user, specify *(Name).value*.

Panel

The *Panel*, similarly to the *Group* component, can be used for structuring the components and for grouping *radio buttons*. The difference between a panel and a group is that the panel does not stand out against the rest of the form as long as it has the same color as the background.

2.9.2 Example

Some examples for the use of dialogs:

Filtering a single system

If the user should only select one particular system, it is best to define a *lookup box* (similar to a drop-down list).

Filtering multiple systems

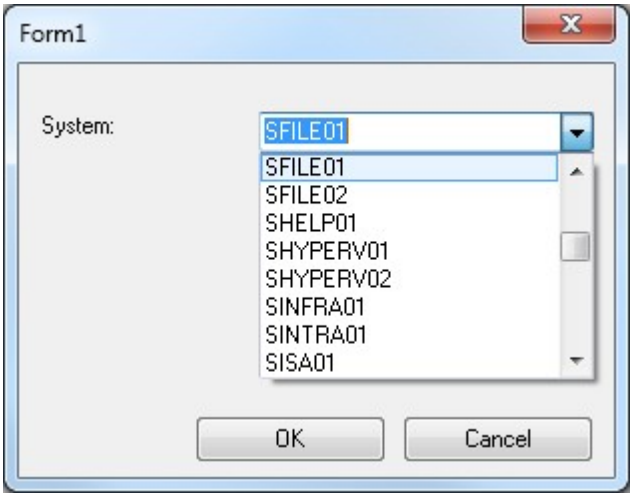
To enable the selection of multiple values at the same time, use a *checked list box*.

Filtering systems using an SQL statement

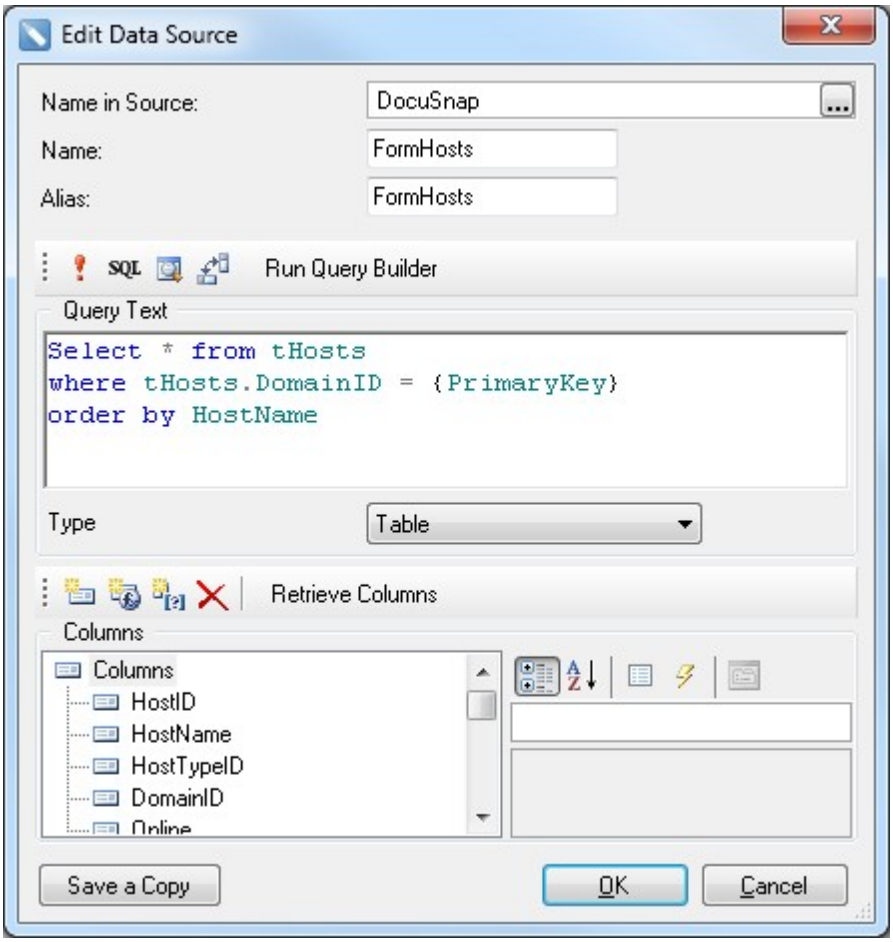
The desired data can be filtered by applying an SQL statement to the data source. The selected values are stored in variables and can then be used in SQL statements.

2.9.2.1 Filtering a Single System

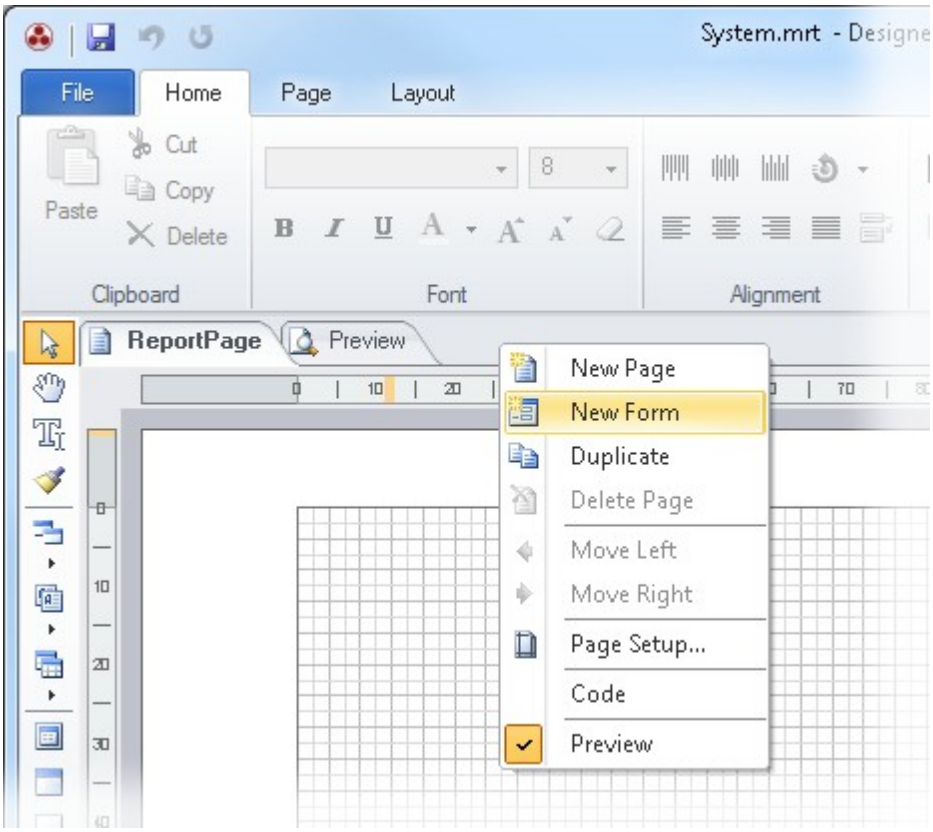
A *lookup box* can be used to enable the selection of a value.



To create this dialog, first define a new data source. This data source can then be used to supply the selection list values. Make sure to indicate in the data source how to sort the list because the components will output the values as they appear in the table without considering the alphabetical sort order.

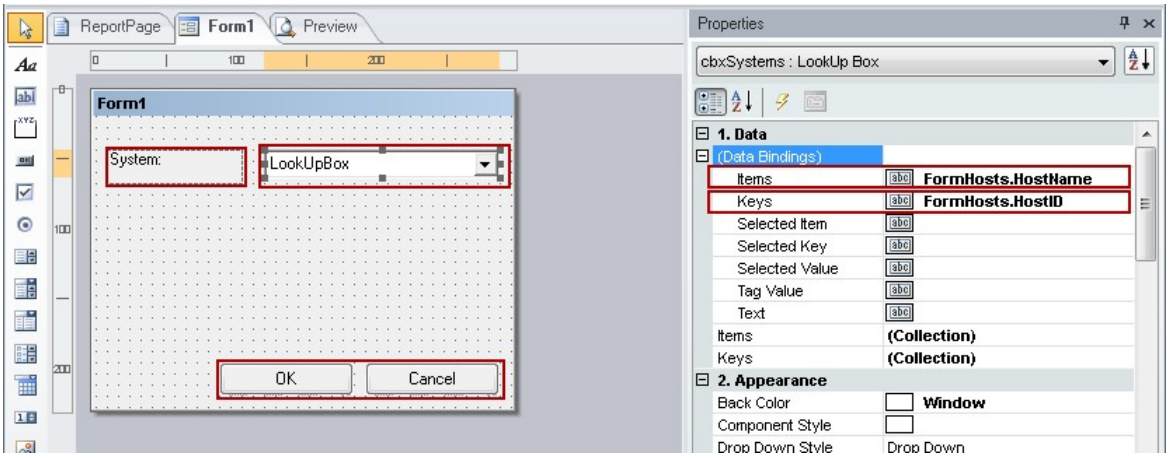


First, create a new form in the Report Designer by right-clicking the horizontal area to the right of the Preview tab and selecting *New Form* from the context menu.



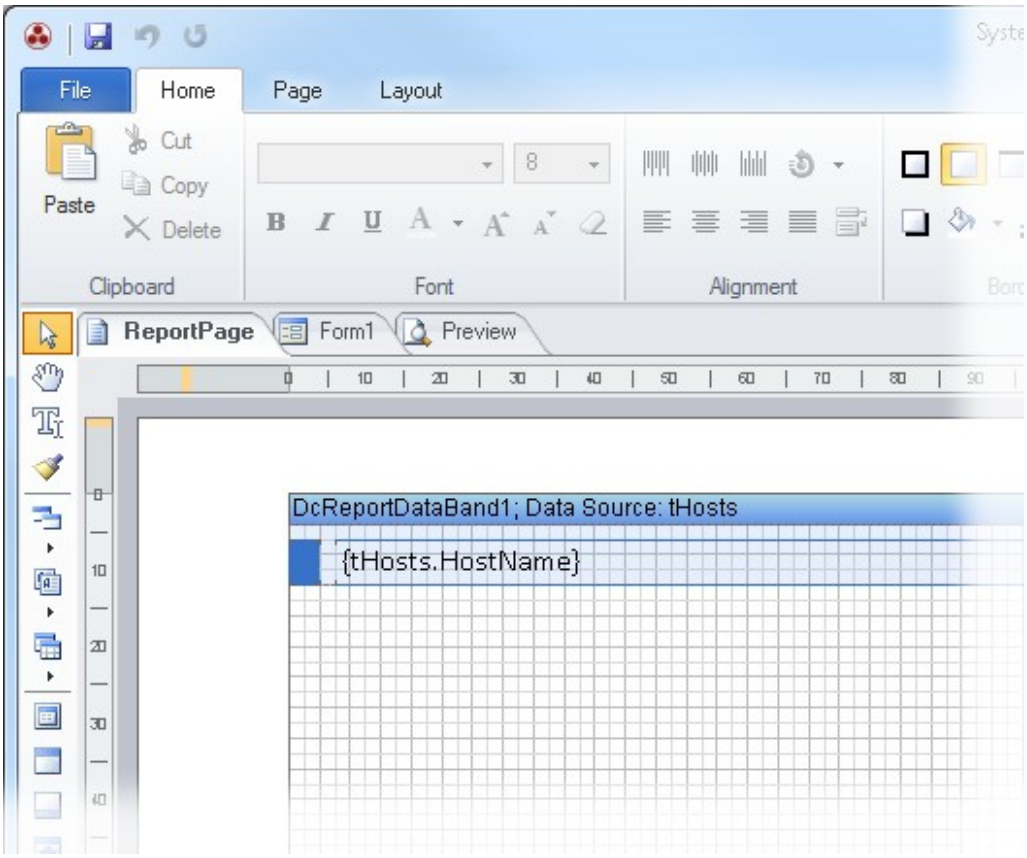
Use the toolbox to add a *label* as the first element and enter *System:* for the Text property.

Then, add a lookup box and select the *FormHosts.HostName* data column for the *(Data Bindings) - Items* property and the *FormHosts.HostID* data column for the *(Data Bindings) - Keys* property. For the lookup box, set the *(Name)* property to *cbxSystems*. Then, add two buttons. For one of the buttons, select *Cancel* as the Dialog Result property, and for the other one, select *OK* as the Dialog Result property. Change the Text properties to *OK* and *Cancel*, respectively. This completes the dialog creation.



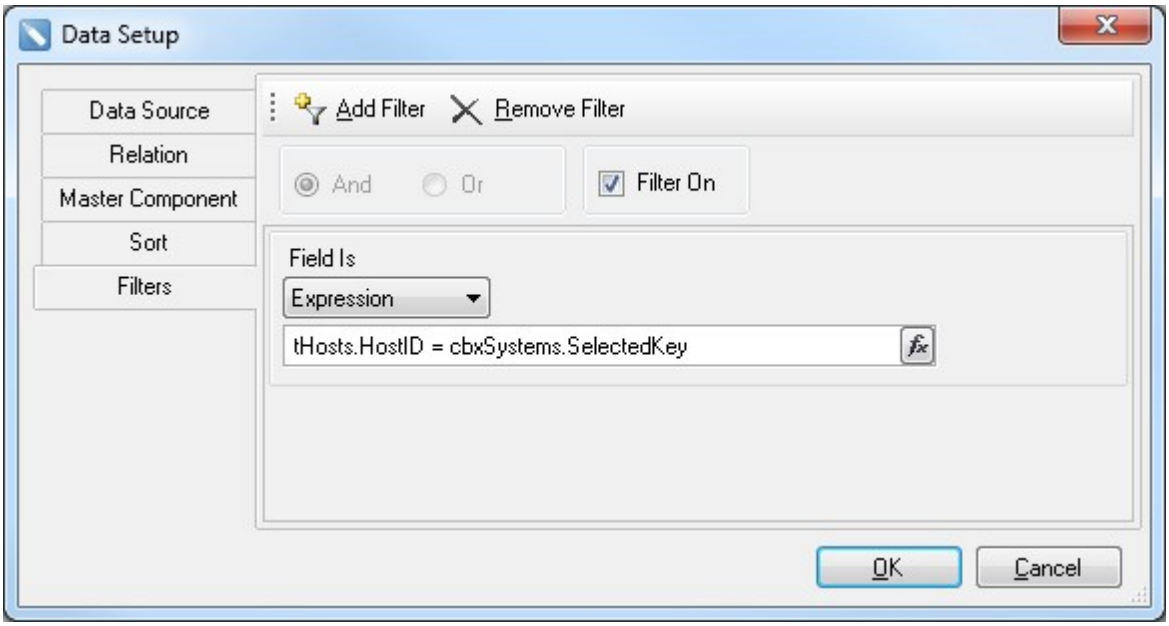
Next, change to the *ReportPage* tab to add the data output components. Define a data band and select the *tHosts* table as its data source. For the table, set the

Connect On Start property to *True*. Then, add a text box to the data band and enter *HostName* as the Text property.

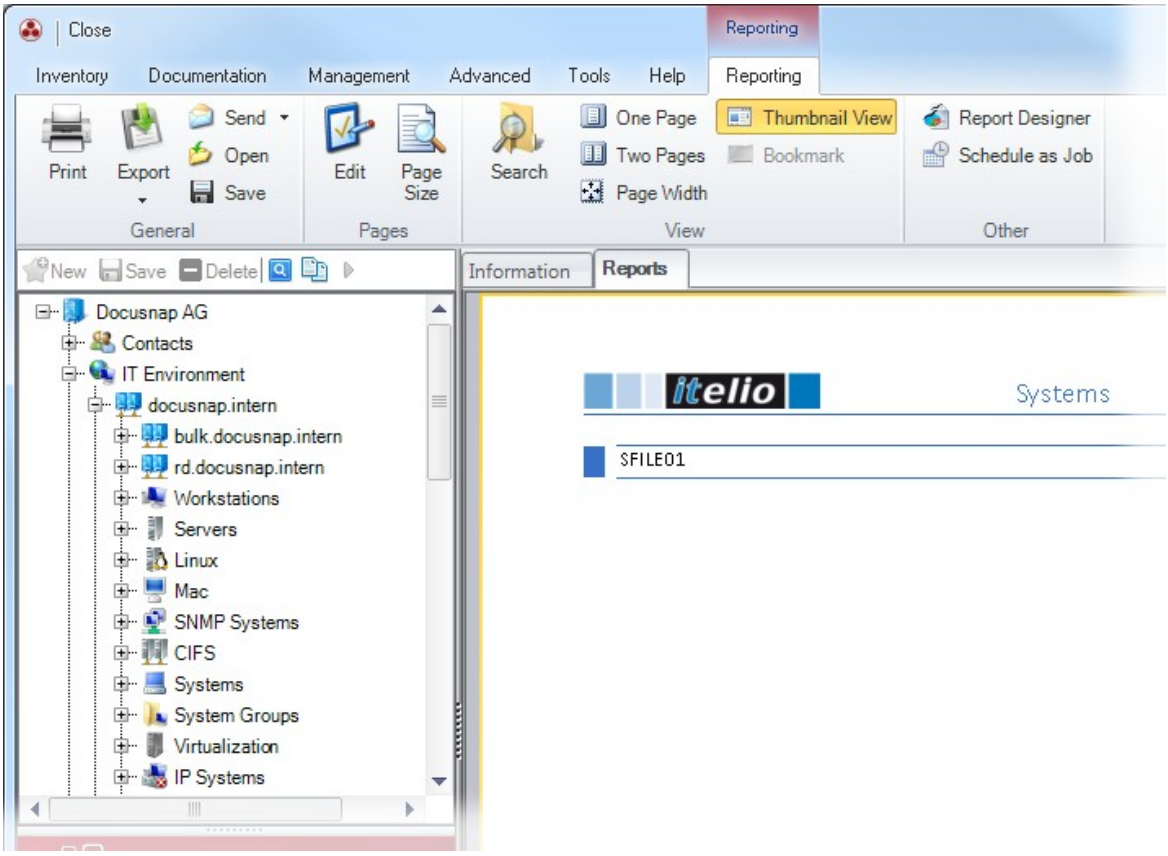


Next, double-click the data band to open the *Data Setup* dialog. Go to the *Filters* page to add a filter and enter the *tHosts.HostID = cbxSystems.SelectedKey* expression for it. This filter determines that only the system whose *HostID* matches the one of the selected system will be displayed.

tHosts.HostID indicates the primary key of the *tHosts* table. The second part of the expression is composed of *cbxSystems* (the name of the lookup box in the dialog) and *SelectedKey* which references the key of the selected value.

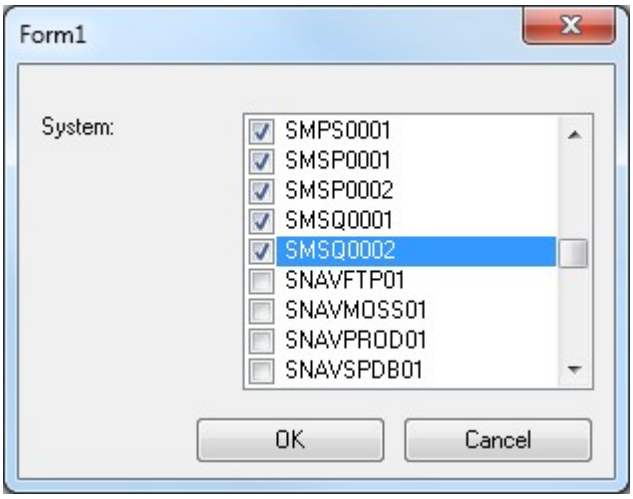


Only the selected value will be included in the report.

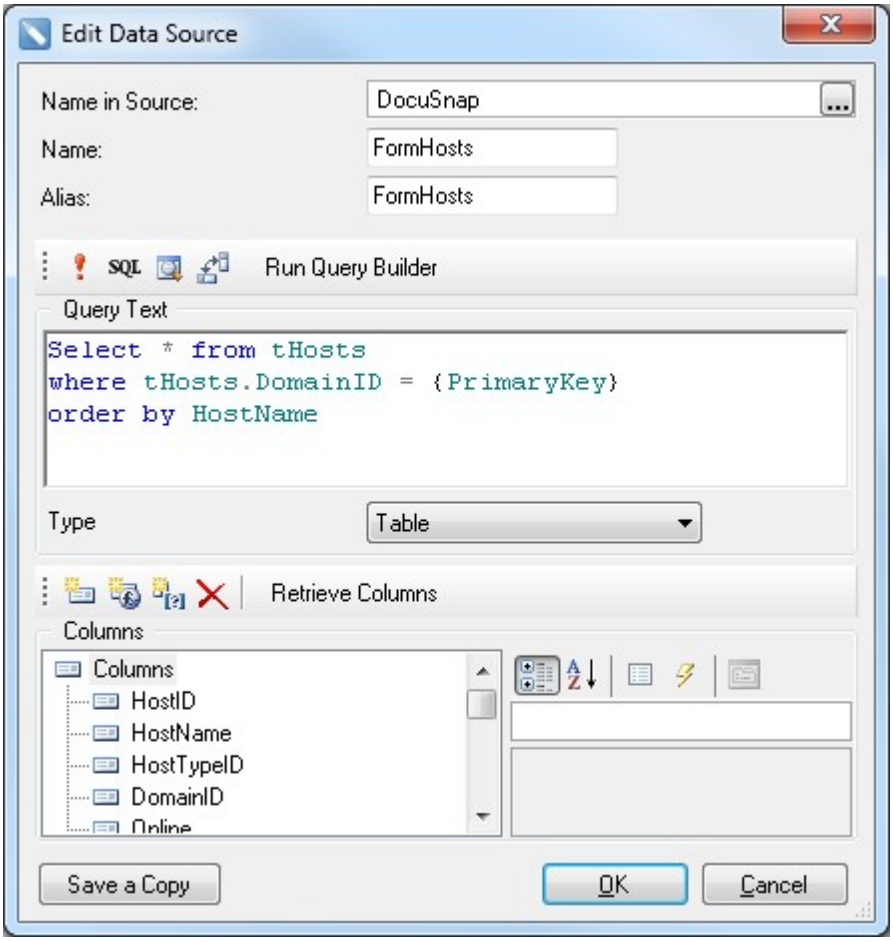


2.9.2.2 Filtering Multiple Systems

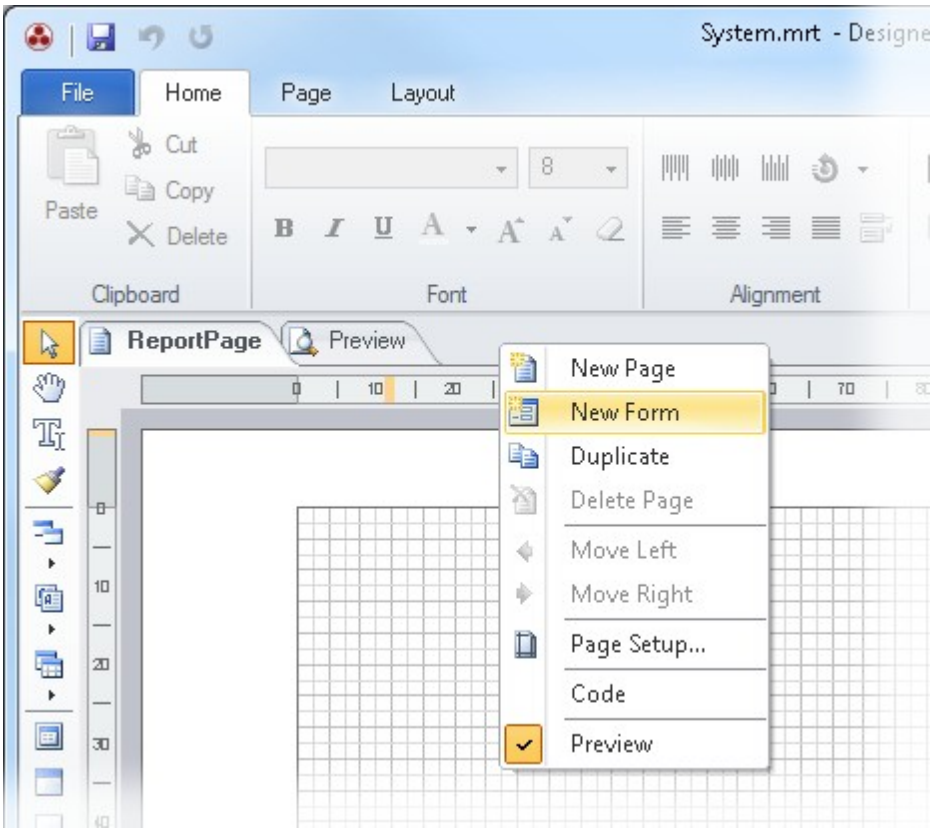
If the user is supposed to select multiple values, it is best to define a *checked list box* for this purpose.



To create this dialog, first define a new data source. This data source can then be used to supply the selection list values. Make sure to indicate in the data source how to sort the list because the components will output the values as they appear in the table without considering the alphabetical sort order.

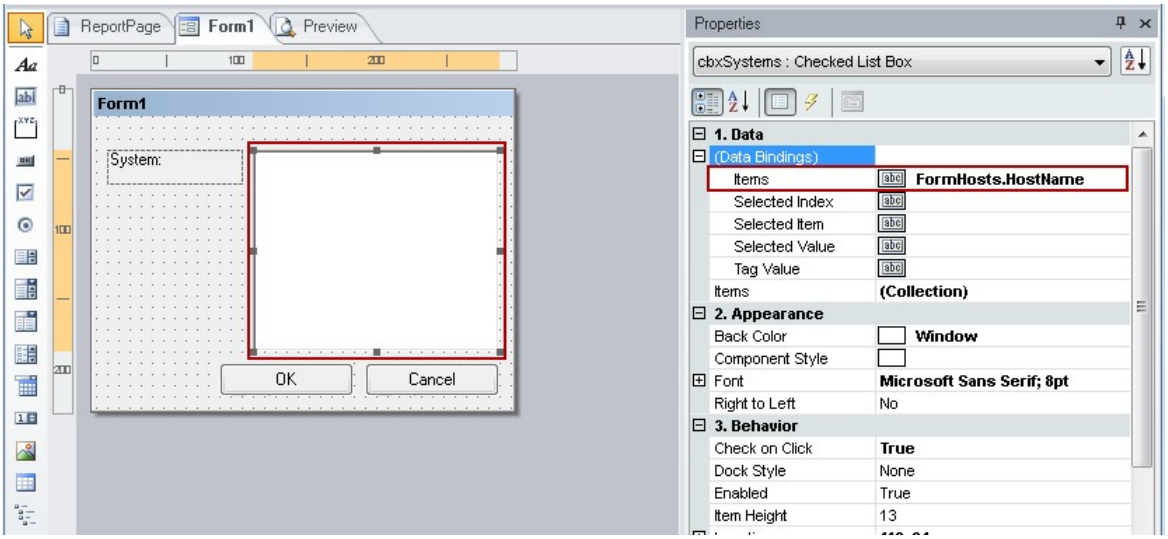


First, create a new form in the Report Designer by right-clicking the horizontal area to the right of the Preview tab and selecting *New Form* from the context menu.

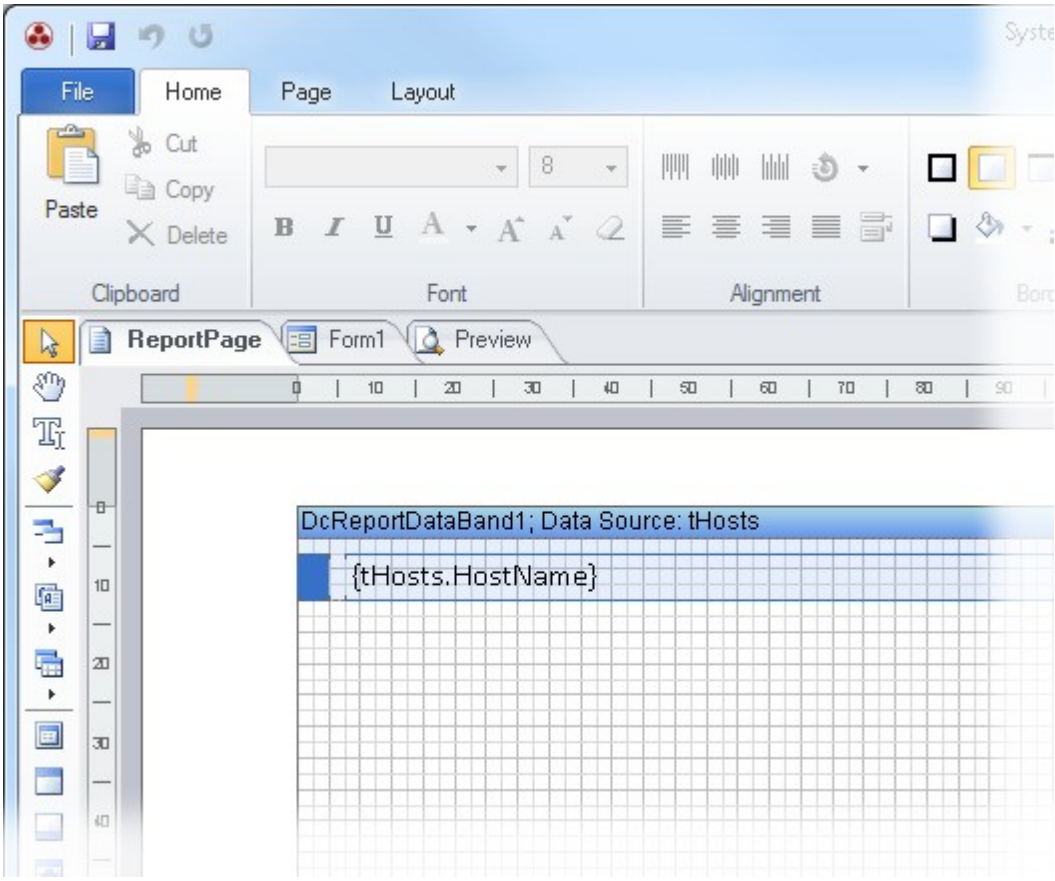


Use the toolbox to add a *label* as the first element and enter *System:* for the Text property.

Now, add a *checked list box* to enable the selection of multiple systems. For the *list box*, specify *FormHosts.HostName* as the data source under the *(Data Bindings - Items)* property. In the *(Name)* property of the *list box*, assign the name *cbxSystems*. Then, add two buttons. For one of the buttons, select *Cancel* as the Dialog Result property, and for the other one, select *OK* as the Dialog Result property. Change the Text properties to *OK* and *Cancel*, respectively. This completes the dialog creation.

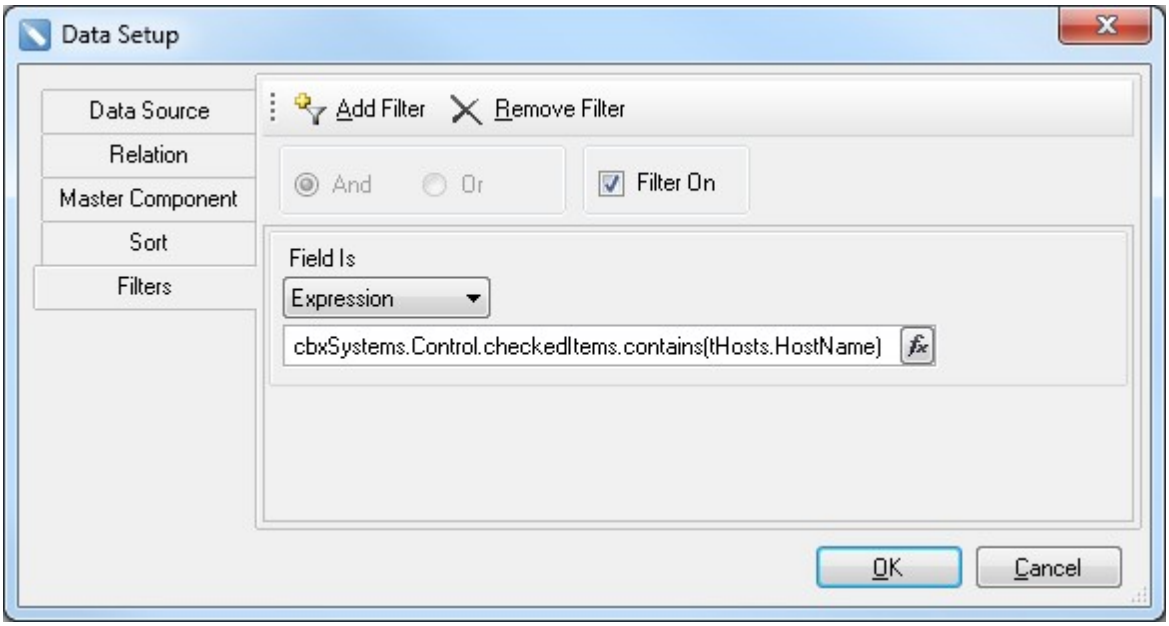


Next, change to the *ReportPage* tab to add the data output components. Define a data band and select the *tHosts* table as its data source. For the table, set the *Connect On Start* property to *True*. Then, add a text box to the data band and enter *HostName* as the Text property.

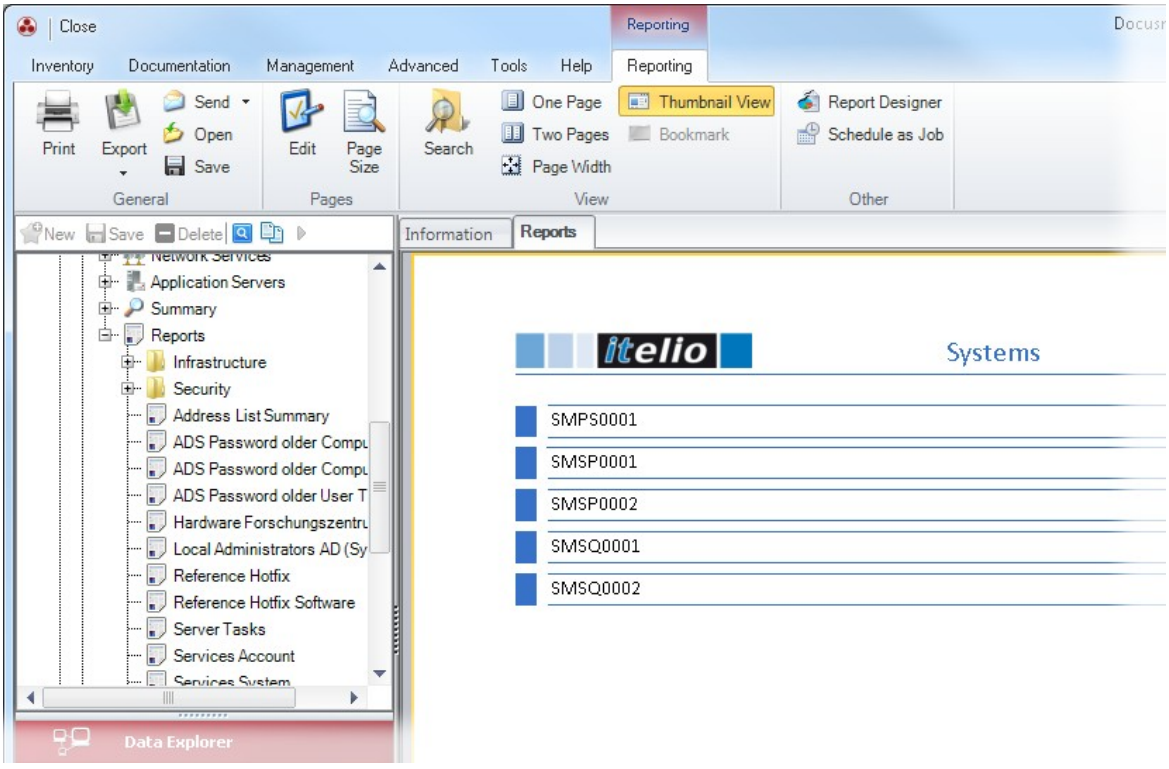


Next, double-click the data band to open the *Data Setup* dialog. There, go to the *Filters* page to add a filter and enter the following expression: *cbxSystems.Control.checkedItems.contains(tHosts.HostName)*. This way, only systems will be output whose name matches the name of one of the selected systems. This statement compares each of the selected systems with *tHosts.HostName* and if it contains this string, the system will be displayed.

The expression is composed of *cbxSystems* (the name of the list box), *Control* to access the control, *CheckedItems* (the list of the selected systems), *Contains* (a comparison function), and *(tHosts.HostName)* which references the names of the systems in the *tHosts* table.



Only the selected values will be included in the report.



2.9.2.3 Filtering Systems Using an SQL Statement

As an alternative to filtering the data using the data band, the desired data can be filtered by applying an SQL statement to the data source. The advantage is that less data needs to be loaded from the database and therefore less memory is used.

This example is based on the report from the Filtering a single system example. To save the selected value from the dialog, right-click in the *Dictionary* panel and select *New Variable*. The New Variable dialog opens. Assign the name

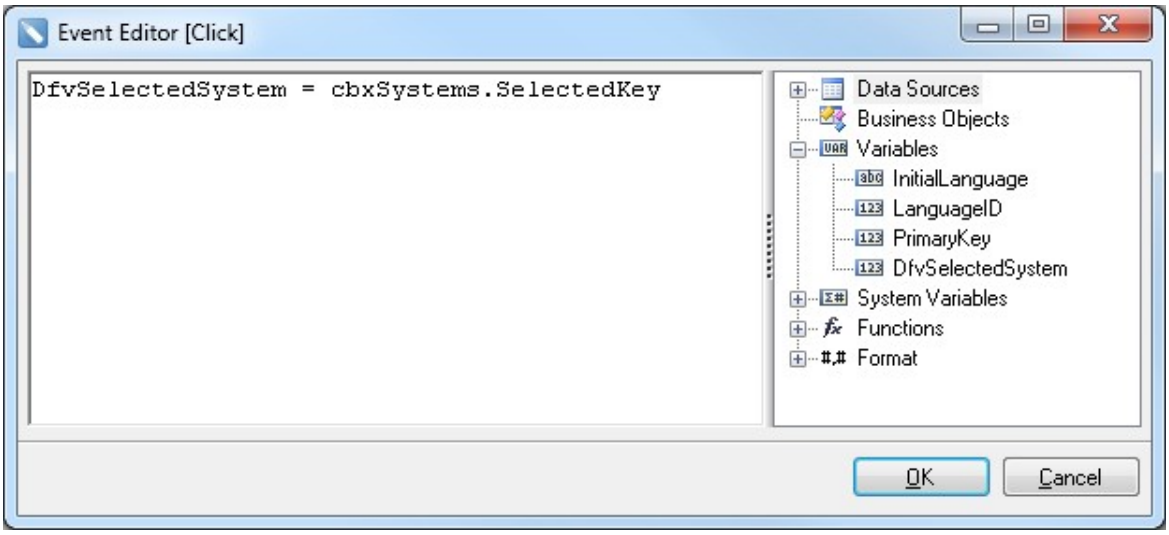
DfvSelectedSystem to this variable.

The *Dfv* prefix is short for *Docusnap Form Variable* and identifies variables which are used for evaluating dialogs. Variables may be named as desired, but if the created reports are to be scheduled using the Docusnap Server, it is recommended to use the *Dfv* prefix to make sure that the values assigned to the variables will be saved.

Now, save a value to the newly created *DfvSelectedSystem* variable. To do so, first double-click the *OK* button in the form. This opens the editor for the *click event*. Enter the following text in this editor:

DfvSelectedSystem = cbxSystems.SelectedKey

This assigns the selected value from the *cbxSystems* lookup box to the newly created *DfvSelectedKey* variable.

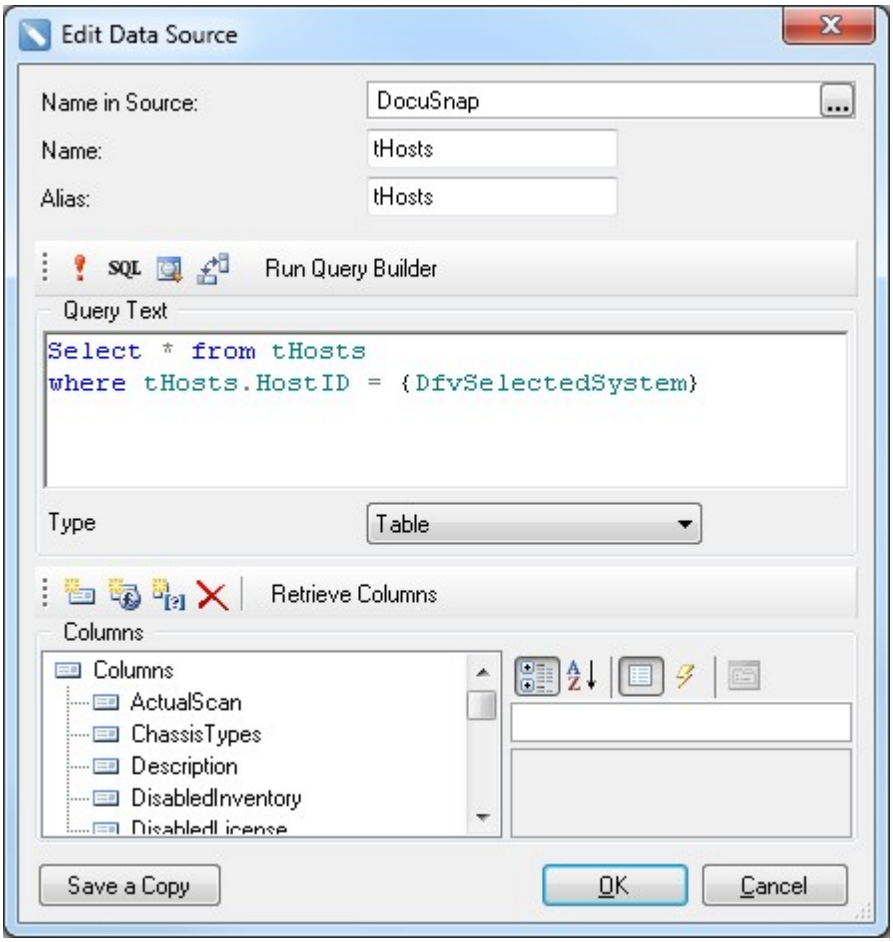


Instead of adding a filter to the data band as described in the Filtering a single system example, you can now use an SQL statement to filter the *tHosts* table on the selected value.

To adapt the SQL statement, right-click the *tHosts* table in the Dictionary panel, and then select *Edit* from the context menu. In the SQL statement, the *Where clause* can be used to filter the *HostID* on the value of the *DfvSelectedSystem* variable:

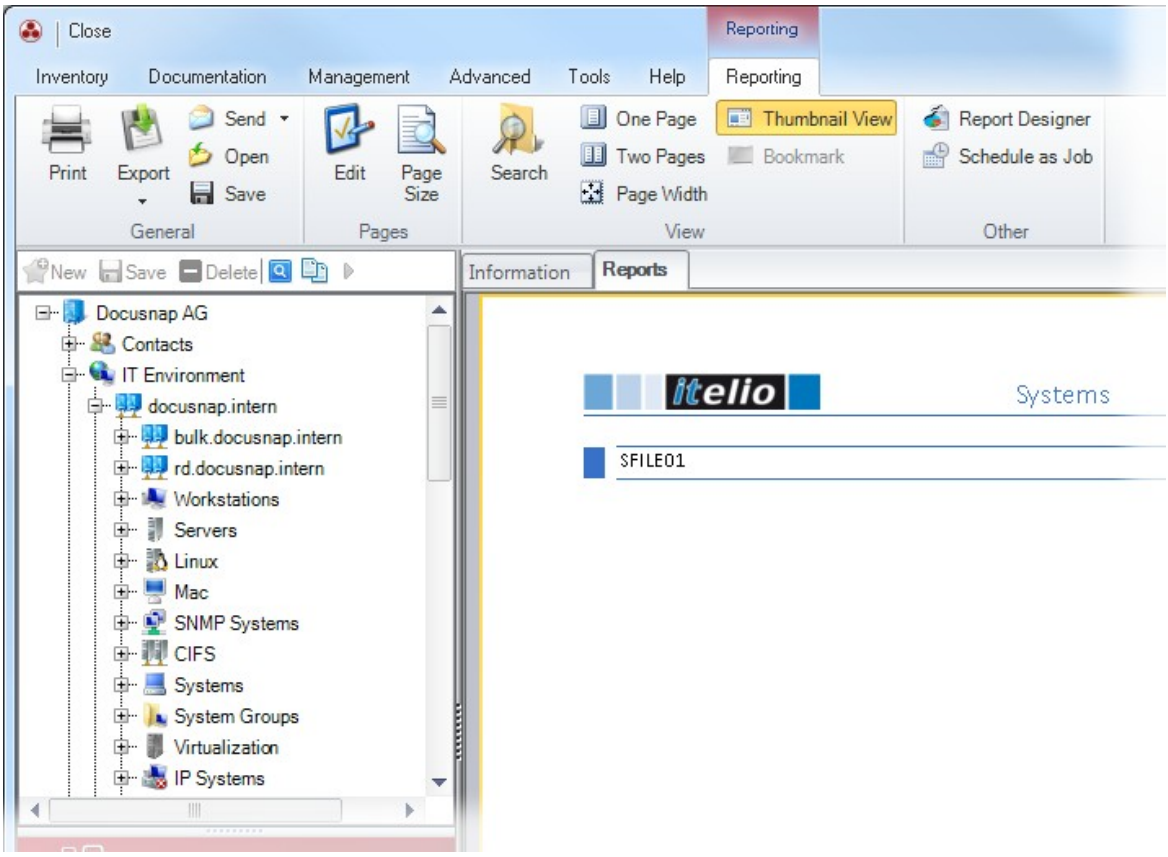
*Select * from tHosts where tHosts.HostID = {DfvSelectedSystem}*

To reference a variable in the SQL statement, enclose its name by braces { }.



The report will then only output the data for the selected system.

Reporting System



Part

3 Database Structures

For storing data, Docusnap uses an open database based on Microsoft SQL Server or Microsoft Access. After the installation of Docusnap and the initial connection to the database, Docusnap automatically creates table structures in the database. These structures are defined in the Metaschema.dss and DDLUpdate.xml files.

If required, additional structures in the database (tables, views and fields) can be created automatically by using the Docusnap Customizing module.

This section consists of two subsections. They include general information on the database and instructions on how to extend the database structure.

3.1 Definitions

In order to simplify working with and understanding the database, certain standards have been used in configuring the database. When extending the database, we recommend to continue using these standards.

All names database elements are in English. For all tables, an auto-incrementing primary key has been defined. The name of the primary key column is always the same as the table name (without the prefix) and has an "ID" suffix. Thus, the primary key column in the *tHosts* table is called *HostID.*

If a foreign key is specified, its name is identical with the name of the primary key column of the linked table. For the *tHosts* table, e.g., a foreign key was defined that links it to the domains in the *tDomains* table. Thus, *DomainID* was specified in the Foreign Key field.

All table names in the Docusnap database begin with a lowercase "t" as an abbreviation for "table". All tables whose name begins with *tSys* do not contain user-definable content. They hold system definitions for Docusnap.

If you create new fields or tables using the Customizing module, they will be identified by an "x" prefix. Thus, you would name a table for mobile phones as "xtMobilePhones", for example. The columns in this table would be identified by a leading "x" (such as xMobilephone). This convention avoids conflicts with system-defined names.

3.2 Organization

Basically, the Docusnap database is structured in a hierarchical manner, starting from the *tAccounts* company table. On the next level, you can find the domains table (*tDomains*) and below it, the child tables for the respective modules.

Thanks to foreign keys, it is easily possible to visualize the database structure in Access or SQL Enterprise Manager.

To find out which tables are used at which level in the tree structures, open the Manage Metaobjects dialog. For more information about this topic, refer to the Meta Objects section.

Tables that contain the word "*docu*" in their name (e.g. *tdocu*, *tADSDocu*), are used to build the individual versions for each type of inventory scan. What is important in this is that each of these tables is linked with the *tSnapshot* table, since each scan is assigned to a unique snapshot. Furthermore, each of these tables contains a field named "*Archiv*" which indicates whether the entry is the most recent one (`Archiv=0`) or belongs to an older version (`Archiv=1`). Thus, using the `Archiv=0` filter criterion is a quick and efficient way to access the most current data without having to spend much time verifying the last inventory scan date.

3.3 Reference Values

To improve performance and to simplify the database for reference lists that contain only a few values (such as Service, StartType, Countries), Docusnap does not use separate tables, but rather reference values.

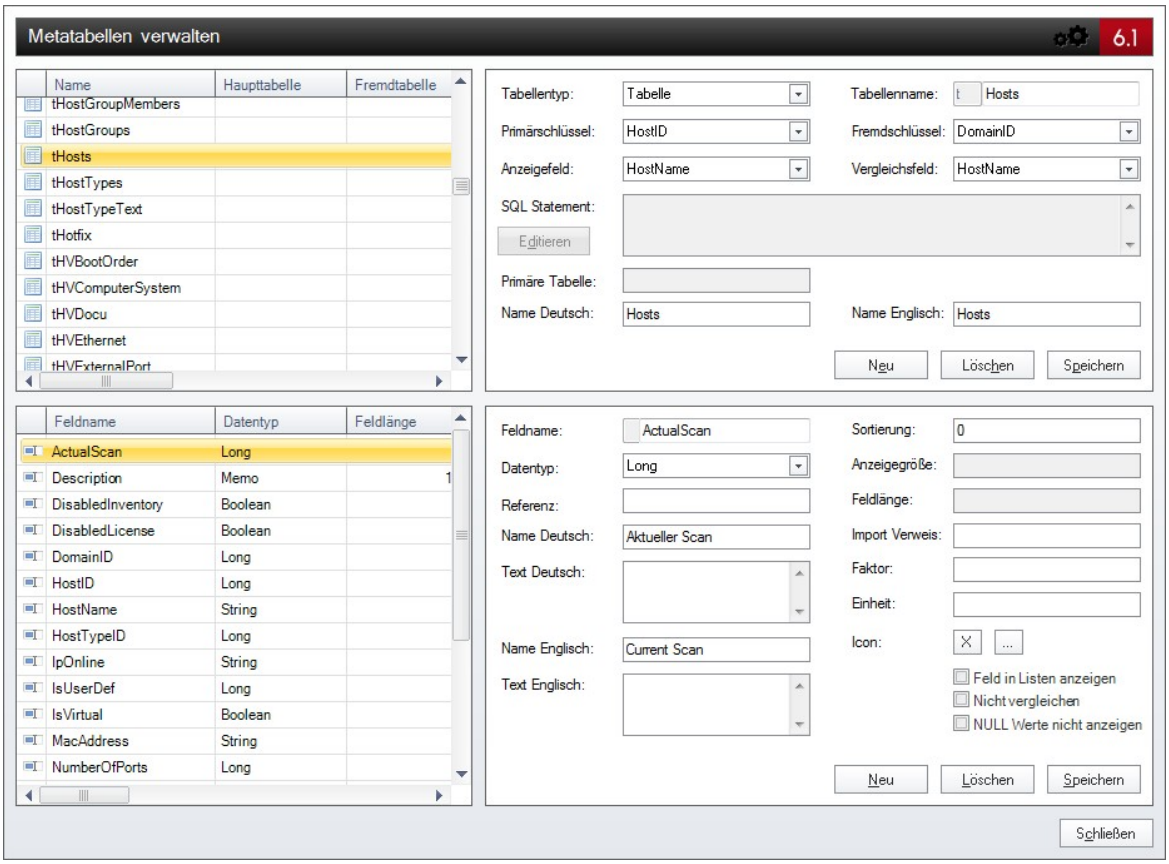
Each reference value consists of a filter value, an ID and the text in German and English, respectively. For a list of the available filter values, see the appendix.

Docusnap controls can either refer to a table or to a reference value. This means that the selection list for a combo box can be populated from either a physical database table or a specific type of reference values.

The defined reference values are stored in the *tSysInitials* table and will be checked for possible changes each time you start Docusnap.

3.4 Modification of the Structure

In Docusnap, you can extend the existing table structures as needed. For this purpose, use the *Manage Metatables* dialog, which can be opened by clicking the *Manage Tables* button on the *Tools* ribbon.



This dialog contains all descriptions of the Docusnap database structure. The available tables are listed in the upper left pane. The corresponding settings for each table can be made in the right pane.

The bottom left pane contains a list of associated data fields, in the bottom right pane, you find the corresponding settings.

In principle, there are three different types of tables in Docusnap:

Table	Tables are the physical Docusnap database tables described in this manual. All table names begin with a lowercase "t" prefix.
View	Views are tables that were generated by an SQL SELECT statement. Views may contain data from multiple tables. It is possible to add placeholders (e.g. FilterID) to the SQL statements. Docusnap will populate them with the current values at runtime. All view names begin with a lowercase "v" prefix.
Virtual	Virtual tables are used to format the tables that are used, for example, in the Docusnap wizards. In addition, they are used in the tree view in order to display data from various database tables in a

	single table. All virtual table names begin with a lowercase "i" prefix.

General Table Properties

When defining a table, you need to set several properties. The number of properties depends on the table type. The following properties apply to all tables, i.e. they are mandatory:

Property	Description
Table Name	Unique name of the table Depending on the selected table type, one of the "i", "v" or "t" prefixes is used. Names of user-defined tables are additionally prefixed with an "x" (e.g. xtSLA).
Primary Key	A field (i.e. column) in the table that uniquely identifies each record. Within Docusnap, these fields are usually auto-increment fields of the LONG data type.
Foreign Key	The field that represents the relation to another table further up in the hierarchy.
Display Field	The field in the table that is used to display the desired text output in the tree views (e.g. Data Explorer, Permission Analysis, etc.).
Comparison Field	For data comparison, a field is required that can be used to identify two records when comparing two snapshots. For this purpose, it might be a good idea to use, for example, a serial number or a computer name.
English Name, German Name	Name of the table in that language.

Note: The *Primary Key, Foreign Key, Display Field* and *Compare Field* fields can only be selected after you have created fields for your table. This means that you must save the table without these

properties first. Then, create the desired fields and finally set the 4 table definition fields

Properties for Views

Views are tables that are built on the basis of an SQL SELECT statement. Docusnap supports this with a number of variables that will be replaced with actual data during the execution of the SELECT statement.

Basically, the SELECT statement has the following syntax:

```
SELECT [DISTINCT] SelectionList FROM Source [WHERE WhereClause]
[GROUP BY (GroupByAttribute)+
 [HAVING HavingClause]]
 [ORDER BY (SortAttribute [ASC|DESC])+];
```

In principle, all valid variants are possible that comply with the SQL standard.

Docusnap provides the following variables for use in statements. Make sure to always enclose the variable in braces, like this: {Variable}.

Variable	Description

{FilterID}	The FilterID variable is always replaced with the primary key of the parent object.
	Example:
	In Docusnap, the following statement:
	`select * from thosts where domainid = {FilterID} order by hostname`
	will result in the following when the statement is executed:
	`select * from thosts where domainid = 1 order by hostname`

{LANGUAGEID}	This variable is replaced with the integer value for the respective language: German = 0 English = 1
{Fieldname}	Using this construct, you can access each data field in a parent data object. Note: This will only work for the Data object type, but not

	for the Caption object type.

Enter the respective SQL statement directly into the *SQL statement* field or click the *Edit* button to open an additional, larger editor window where you can enter or edit the statement.

Since an SQL SELECT statement may be used to link multiple tables, Docusnap does not know which table the primary key refers to when you enter a DELETE statement. In the *Primary Table* field, you can specify the table in which the record should be deleted. If you leave this field blank, it will not be possible to delete meta objects that depend on this table.

SNMP Statements

In addition to SQL statements for the output of tables, Docusnap provides special statements for the output of SNMP tables. When performing an SNMP inventory scan, Docusnap uses MIBs. To program the output, enter the following into the *SQL Statement* field: First, the text *SNMP:*, then, in parentheses, the MIB and, separated by a comma, its name. To separate the MIBs, enter a semi-colon.

This results in the following syntax:

```
SNMP:(1.3.6.1.2.1.4.20.1.1,IPAddress;1.3.6.1.2.1.4.20.1.3,
SubnetMask;1.3.6.1.2.1.4.20.1.2,InterfaceID)
```

To ensure the output of values that exist only once for each SNMP device, precede the statement with SNMP-Single.

Use the following syntax:

```
SNMP-Single: (1.3.6.1.2.1.43.8.2.1.14.1.1,Manufacturer;1.3.6.1.2.1.43.5.1.1.17.1,
SerialNumber)
```

If you need to divide the value of one column by that of another column, e.g. to determine the toner level, separate the MIBs of the two columns using a slash. When the statement is executed, the quotient will be output in this column.

The statement has the following syntax:

```
SNMP-Single: (1.3.6.1.2.1.43.11.1.1.9.1.1/1.3.6.1.2.1.43.11.1.1.8.1.1,TonerBlack)
```

Then, create the fields (columns) of this table. As the field name, use the name you entered in the statement for this MIB.

Data Fields

Data fields represent the various columns of a table. Each field is assigned a data type in Docusnap. The meta description of each field also includes its English or German name.

Property	Description
Field Name	In this field, you can specify a unique field name. Key fields should always have the "ID" ending, so that they can be identified more easily. User-defined fields will automatically be assigned an "x" prefix to avoid naming collisions.
Datatype	This field indicates the data type of the field. Blob — Binary field that stores binary data, such as attachments Boolean — Yes or No value Byte — Numeric field that holds integer values between 0 and 255. Date — Date field Decimal — Decimal field for floating point numbers Combination — Field that combines multiple fields. Field names are identified by ampersand "@" symbols, literal text is surrounded by single quotes ('Text'). The values can be concatenated using the "+" operator. Strings in angle brackets "<>" are only displayed if they are followed by more text. Example `@Lastname + '<, >' + @ Firstname`

	Output: Smith, John or, if no first name exists, Smith The Combination type should only be used for display fields. Long — Integer Memo — Text box containing an unlimited string of characters String — Text
Reference	By means of the *Reference* field, an ID can be converted into a meaningful (plain text) name. For this purpose, enter a reference to a table [TableName] or a reference to a reference value {ValueName}. For a listing of possible reference values, see the list in the appendix. If you have specified a value in the Reference field, Docusnap uses the numeric content of the entry and performs a query on the specified table or in the reference values, filtering on the respective primary key. The output will be the value that was defined as the display field in the target table. Example: Field: DomainID, Reference: [tdomains] Instead of the DomainID "1", the output will be "test.local".
Name, Description (German, English)	Field name in the respective language. This value will be shown later in lists as a column heading.
Sort Order	The field order in lists is not alphabetical, but rather determined by this value.

Display Size	Here, you can specify the column width (in pixels) for the display in lists. If you do not specify a value, Docusnap uses a default value of 80 pixels.
Field Length	This value can only be defined for the *String* data type. It is used to enter the maximum number of characters that may be entered in the respective field.
Import Link	This field defines reference fields that must be compared and matched for data import. Basically, these are references such as comment types, passwords types, etc.
Factor	In this field, you can set a divisor for converting numeric values. This field is only enabled for numeric data types. For example, a factor of 1024 would convert a byte value into kilobytes.
Formatting Template	Using this field, you can define the formatting of numeric values according to standard conventions. Valid placeholders include the hash symbol "#" for any numeric value and "0" for numeric values that should be padded with leading zeroes if they are too small for the field. The thousands separator is a comma, and the period is used as the decimal separator. Examples: #,##0.00 MB 00 h
Icon	Using the Icon property, you can store a specific icon for each field. This is only necessary if the meta object that refers to the table has a vertical alignment.
Field Visible in Listings	If you tick this checkbox, this field will be visible in lists.
Do Not Compare	By enabling this checkbox, you can prevent the field from

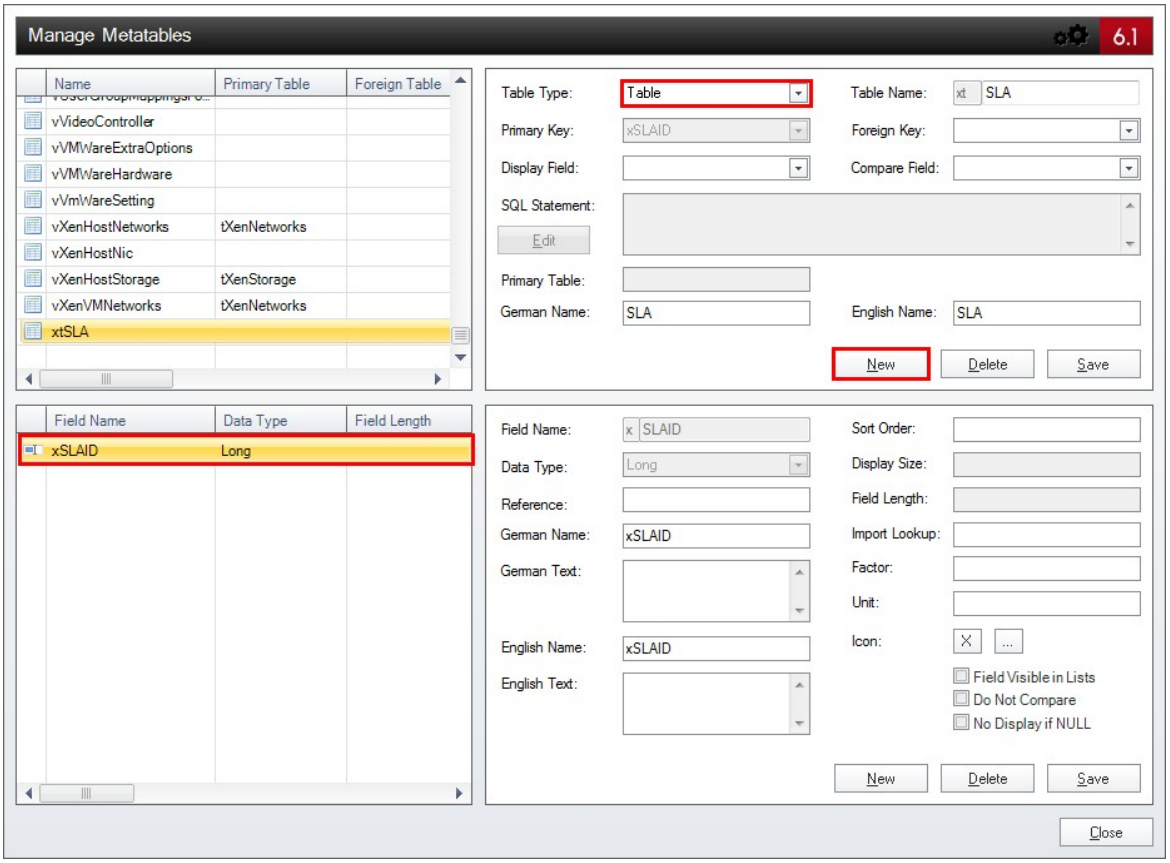
	being used in data comparisons. This is useful, e.g., for a field that reflects the available storage space on a hard disk, since this value is ever-changing.
No Display if NULL	If the meta object for which the table will be shown has a *Vertical* alignment, this flag determines whether the column will be displayed at all if the database value is NULL.

3.5 Example

As a simple example, this section explains how to create an additional table for entering SLAs (Service Level Agreements).

Creating the Table

The first step is to create the table. To do so, click the *New* button. Then, select *Table* as the table type and enter the table name, i.e. *SLA*. Save the table by clicking the *Save* button. The xSLAID column is created automatically and defined as the primary key.



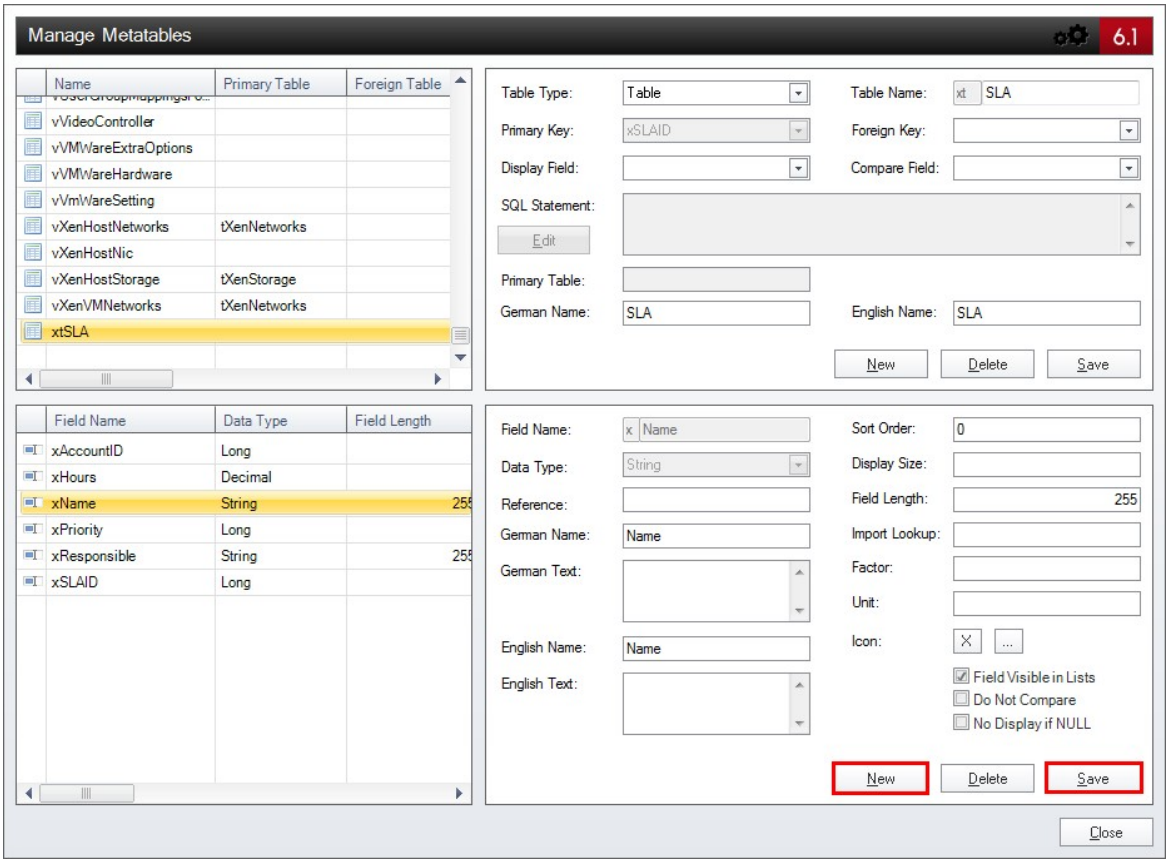
Creating Fields

Next, click the *New* button in the lower pane to create additional fields. To save a newly defined field, click the *Save* button in the field creation pane.

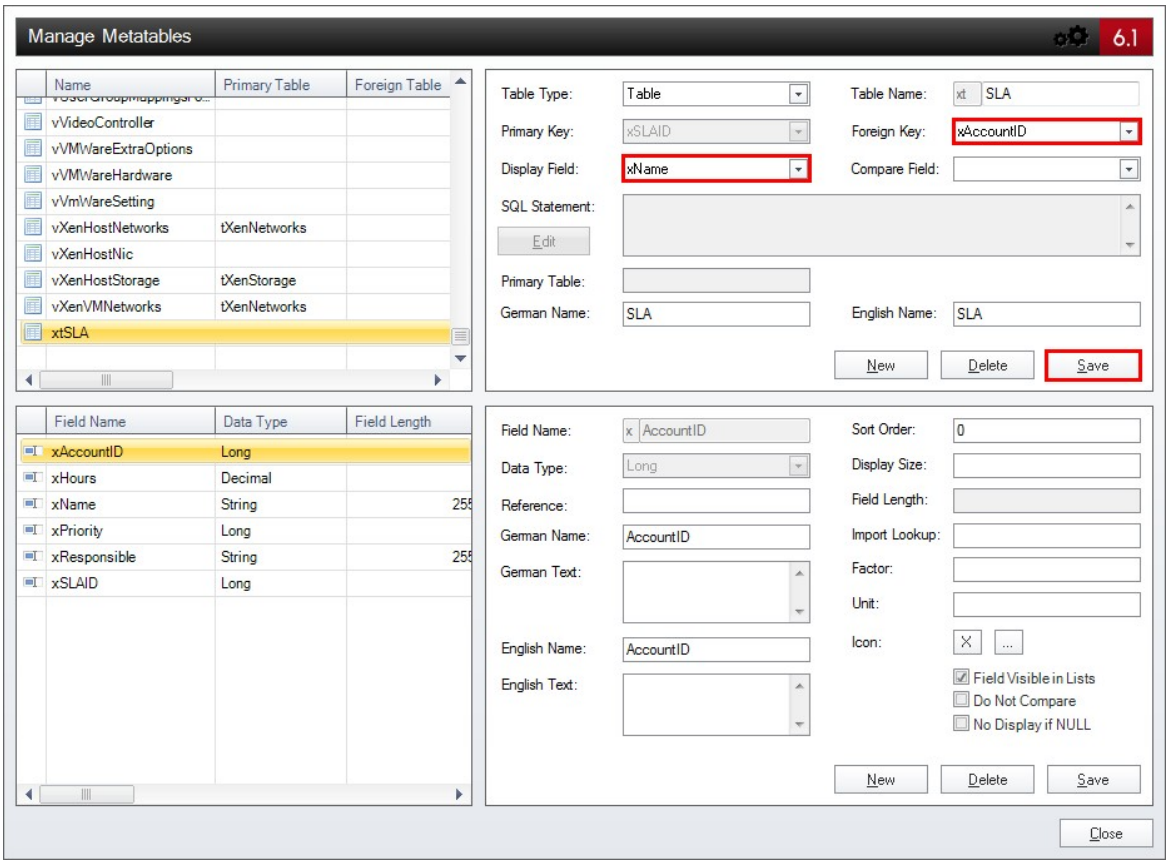
For this example, you need to create the following fields:

Field Name	Datatype	
AccountID	Long	
Hours	Decimal	
Name	String	Field Length: 255
Priority	Long	Reference: {RePriority}
Responsible	String	Field Length: 255

When you are done creating these fields, select the *xAccountID* field as the foreign key and the *xName* field as the display field. When you have selected the foreign key and the display field, click the *Save* button in the upper pane to save your changes.



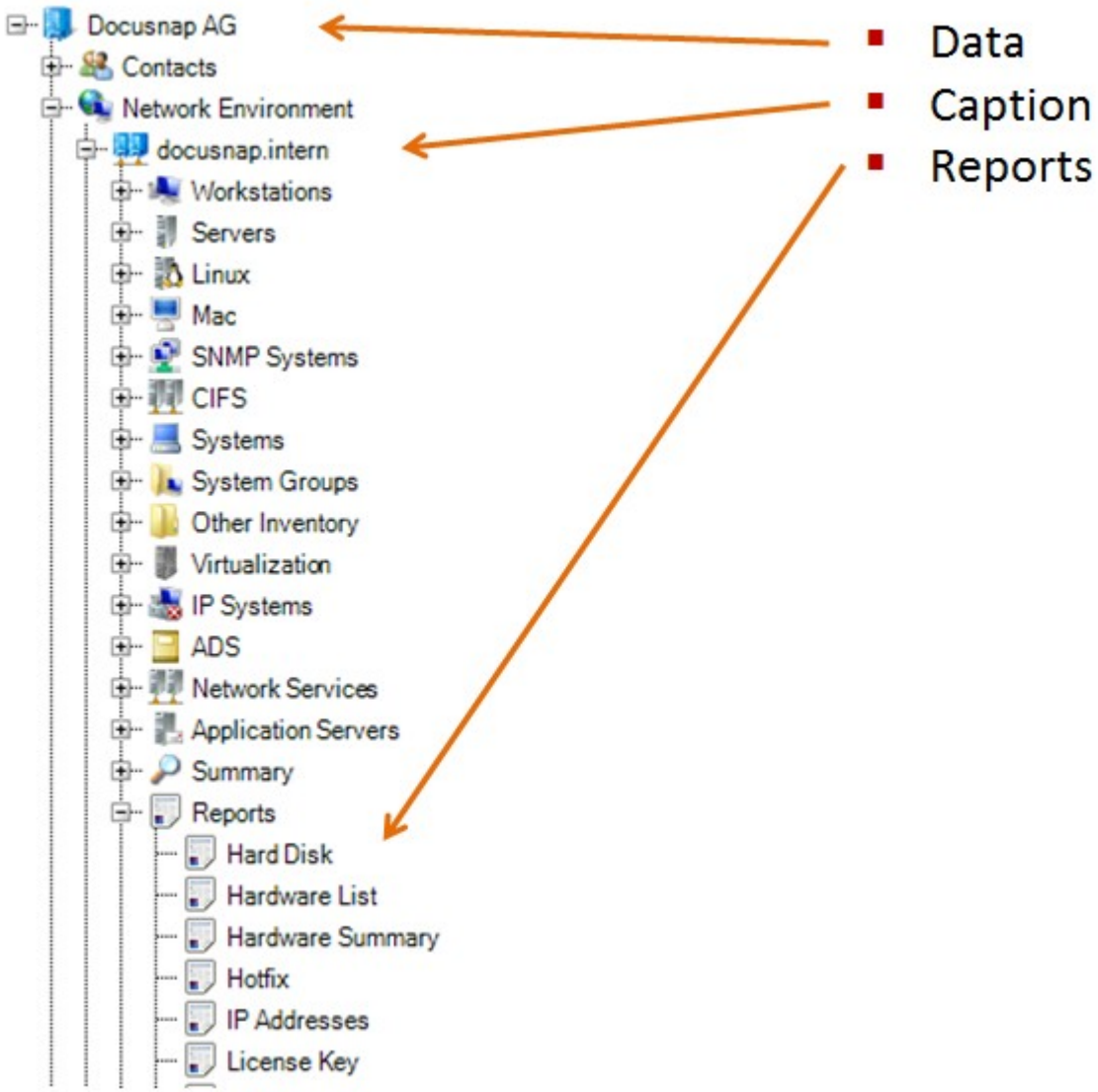
Database Structures



Part

4 Meta Objects

The Docusnap tree structures consist of meta objects that can be extended as desired. In total, six different tree structures are available in Docusnap (Data Explorer, Permission Analysis, License Management, Organization, IT Documentation and Data Import (not visible)).

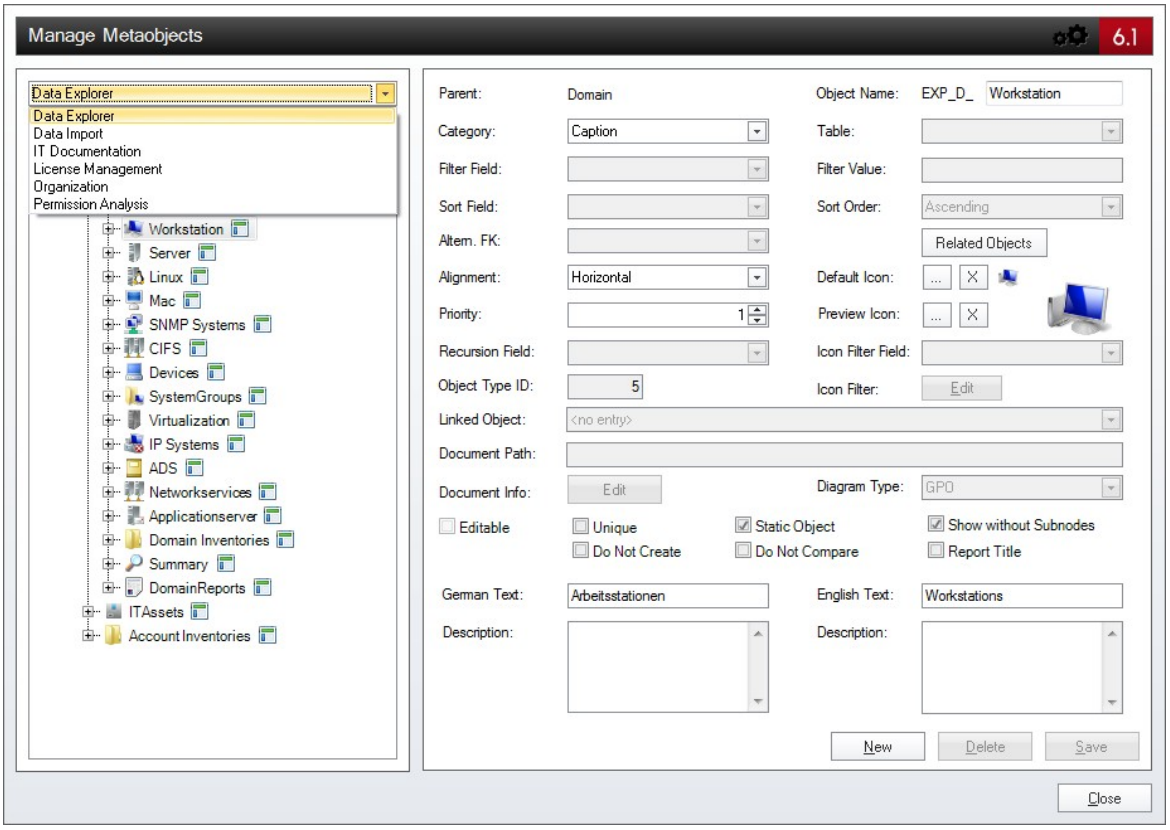


Meta objects are the items underlying the individual tree nodes. A node can belong to one of the following categories: Caption, Data, Report, Linked Object, Output or Diagram. Meta objects define the contents and organization of the tree structure. Meta objects of the Data and Report types are placeholders for the corresponding data.

4.1 Basics

In the *Manage Metaobjects* dialog, you can adjust the tree structure. This dialog

provides all settings required to create user-defined structures.



In Docusnap, five different hierarchical structures are available that can be used to organize the data that has been recorded.

Hierarchy	Content
Data Explorer	All data retrieved by a scan or entered manually.
Data Import	This hierarchy determines the structure for importing data from other databases.
License Management	Represents the hierarchy for the License Management module.
Organization	Hierarchy for the Docusnap Organization module.
Permission Analysis	Hierarchy for the Permission Analysis module.
IT Documentation	Structure for the IT Documentation view.

You can extend all hierarchies by means of the *Manage Metaobjects* dialog. Certain properties and settings apply to all meta object types. They will be explained below. The specific properties of the various object types will covered in separate subsections.

For each meta object, a unique primary key is stored at runtime that allows its identification. Each object without a primary key (such as captions) will inherit the primary key of its parent Data-type meta object.

In simple words, each child level is filtered on a value from its parent level. Thus, the domain level is filtered on the value of the AccountID column, which represents a unique value at the company level.

General Settings for Manage Meta Objects Dialog

Property	Description
Object Name	This field contains the name of the object. The object name is composed of a prefix + its origin, i.e. defined by the software manufacturer ("_D_") or by the customer ("_U_") + the object name. The prefix indicates in which tree hierarchy this meta object is located. The following prefixes are available: ▪ Data Explorer (EXP) ▪ Data Import (IMP) ▪ License Management (LIC) ▪ Organization (ORG) ▪ Permission Analysis (RIG) ▪ IT Concepts (CPT) Example: A new object for a caption (such as Service Level Agreement) would have the following name: EXP_U_SLA
Category	This field indicates the object category or type (Caption, Data, Report, Linked Object, Output or Diagram).
German Text / English Text	The display name of the object in English or German, respectively (only visible for captions in the tree)
Priority	If you create multiple meta objects at the same level, their order

	is not alphabetic, but determined by the number in this field. Thus, the Server meta object under a domain, for example, has a higher number than the Workstations meta object.
Alignment	This field indicates how the results from the list in the right Docusnap pane will be aligned.
Object Type ID	Unique ID of each meta object. Docusnap assigns these IDs automatically. By default, all IDs below 1000000 are reserved for system definitions.
Default Icon	Defines the icon that is displayed next to the object in the tree.
Preview Icon	Defines the icon to be used for the creation of diagrams that illustrate IT relations.
Document Path	Using this property, you can link external documents, such as Word or HTML files, with this meta object. When you select an object, Docusnap automatically checks whether documents exist in this path. If documents are found, they will be displayed on the separate *Documents* tab of the Docusnap Data pane. The path entered for this property is always relative to the documentation directory. In order to make paths flexible, you can use variables when specifying the path. The following variables are available: %Account% Company name %Domain% Domain name %Object% Object name %ParentObject% Object name of the parent node Docusnap will automatically append "_EN.html" or "_DE.html" to the file names. Example:

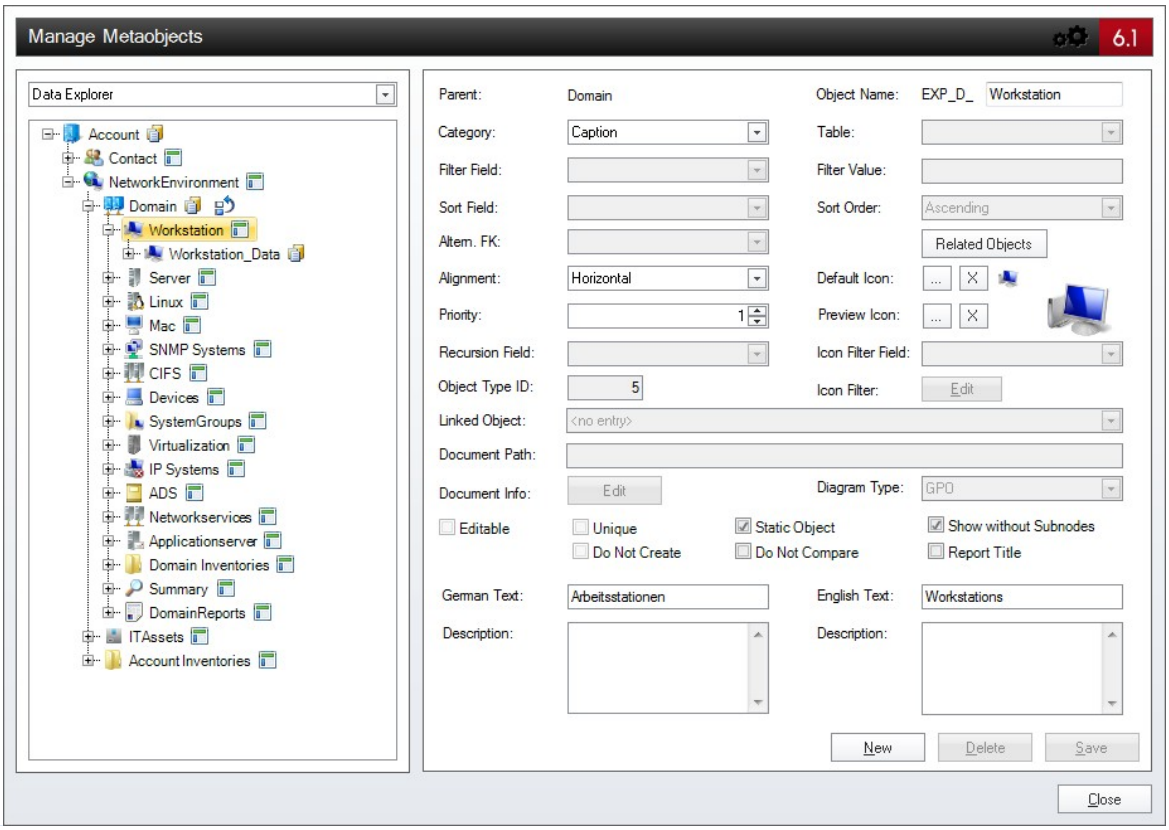
	Assuming your documentation directory is "c:\Documentation" and you enter %Account%\%Domain%\Datasheets\Workstations\%Object%\%Object% in this field, the resulting file name will be: c:\Documentation\Docusnap AG\test.local\Datasheets\Workstations\Notebook1\Notebook1_en.html
Show Without Subnodes	Objects of this meta object type will even be displayed if they do not have any child nodes. By default, Docusnap hides objects without child nodes, i.e. this checkbox is disabled. Enabling this checkbox also has an performance increasing effect, because Docusnap will not have to check whether there are subnodes or not.
Do Not Compare	The Docusnap data comparison will ignore objects for which this checkbox is enabled. Objects at levels below this one will neither be compared.
Report Title	When you generate/execute a report, the first object that is higher in the hierarchy and for which this checkbox is enabled, will be used as the report subtitle.
Static Object	This option enables the extensions (comments, passwords, etc.) for this meta object.

If the alignment is set to Vertical, make sure that no more than one record is returned for each level. Otherwise, the data output will fail.

For a vertical alignment, Docusnap will ignore the default icon that has been specified. In this case, specify the icon to be used for each individual field from the Manage Metatables dialog.

4.2 Captions

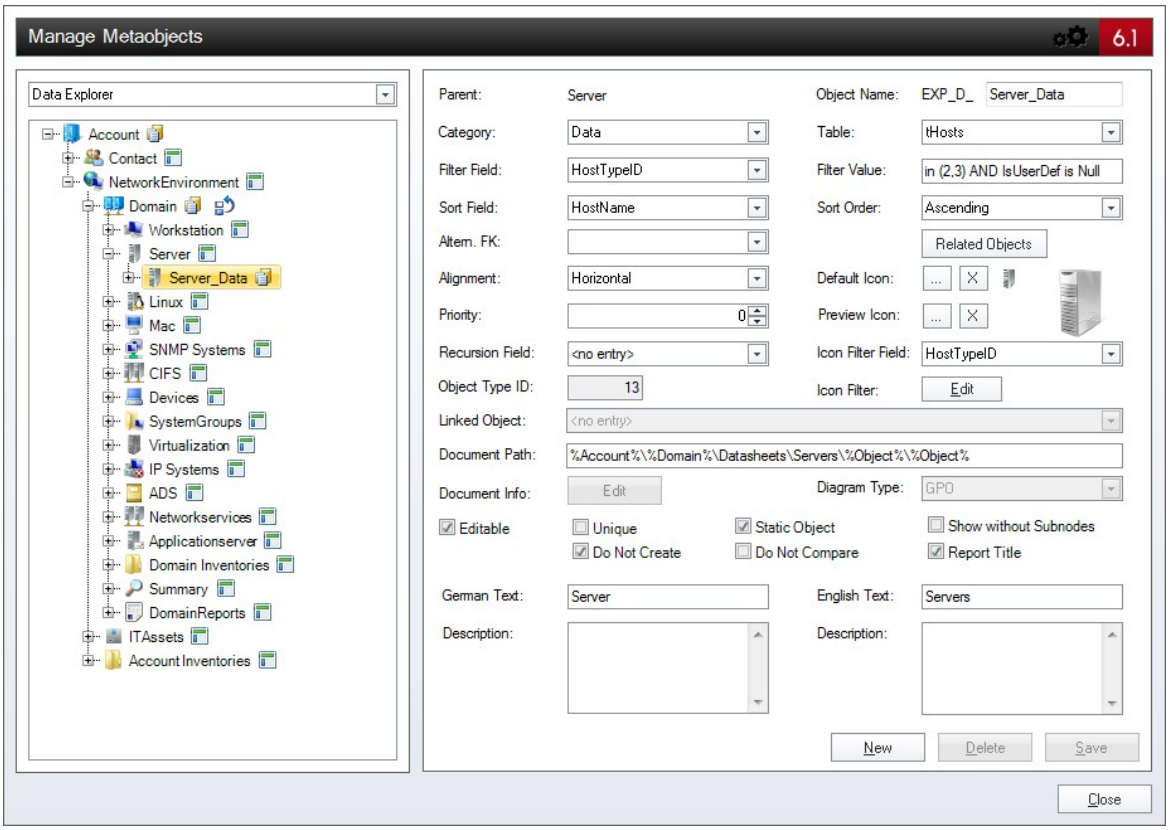
Captions are not linked with any table. They are used to organize the objects and therefore help to present the tree structure more clearly.



For a meta object of the Caption type, all general settings are available.

4.3 Data

A meta object of the Data type is used for the output of database tables in the hierarchy. For this reason, such a meta object must be linked with a database table, view or virtual table. For this object type, additional settings and options, especially for the display of content data, are available.



Virtual tables are used for the output of data from various tables in a single table where they can be sorted by a specific value. A virtual table node will not be shown in the tree view, but rather is used to combine the data from the dependent tables. The column names in the virtual table must match the field names in the tables being combined. For this reason, the tables to be used here should share some columns. The primary key of the parent node is used as the a foreign key for the subnode. In order to show different icons in the table for display in the main window, define such icons using the *Icon Filter Field*.

Available Options for Data-Type Meta Objects

Property	Description
Table	Here, you can select the linked data table from the list of meta tables. Meta tables can be modified or extended from the *Manage Metatables* dialog.
Filter Field	This is the table field used for defining a filter.
Filter Value	Filter criterion for the *Filter Field*. In this field, you can enter any valid SQL conditions (such as `= 1`). The clause can further be extended using additional fields or conditions.

	Examples:
	`= 1 AND Hostname Like 'S%'`
	`<> 5 AND HostType in (1,2,3)`

Sort Field	Database field on which the displayed elements will be sorted.
Sort Order	Order in which the data of the sort field will be sorted.
Alt. FK	An alternate foreign key is used if you want to build the hierarchy using a foreign key that is different from the one specified in the *Manage Metatables* dialog. This option can only be used with real tables. If a data entry screen is used for this node, the alternate foreign key column of new entries will automatically be filled with the primary key of the parent node. The actual foreign key for the table must be entered using the data entry screen.
Editable	If this checkbox is enabled for a meta object, a data entry screen can be created or is available..
Unique	This flag ensures that only one object can be created for each level. This option is can only be selected for meta objects where the *Editable* checkbox is enabled.
Do Not Create	If you tick this checkbox, the object can only be deleted or saved. The *New* button is disabled in this case. This means that users can only edit objects, but cannot create new ones
Recursion Field	By means of this property, you can define a recursive field, and thus build a hierarchy on the basis of a table. At the first level, only records will be shown where the recursive field contains the value NULL or -1. Make sure to always define a recursive field as a Long data type. At the second level, Docusnap shows records where the recursive field contains the value of the primary key from the first level.

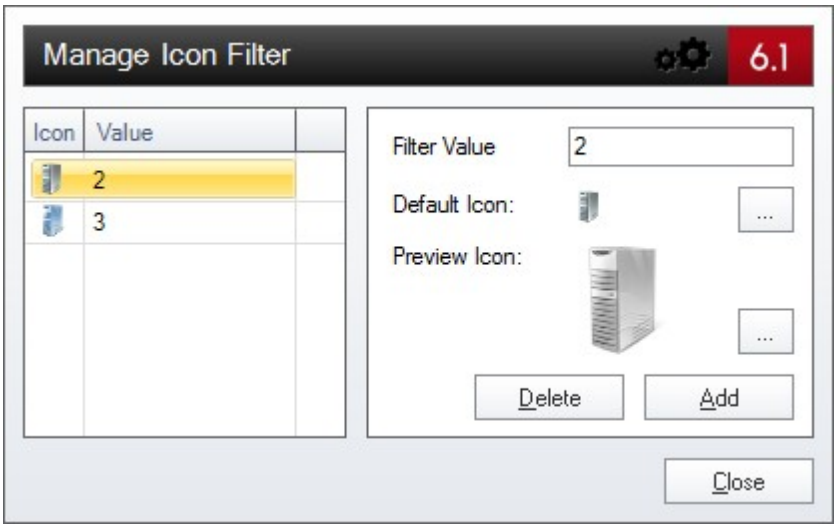
Icon Filter

Using an icon filter, you can assign one of various icons to a meta object based on

the value of a certain column. Thus, it is e.g. possible to define only a single meta object to be used for workstations, servers and domain controllers, but assign different icons by means of the icon filter.

To enable the icon filter, select the data field that contains the filter value from the *Icon Filter Field* field.

Click the *Edit* button to open a separate dialog where you can define the desired filter values and the associated icons. If the table includes a value for which no custom icon has been defined, the default icon for that object will be used.

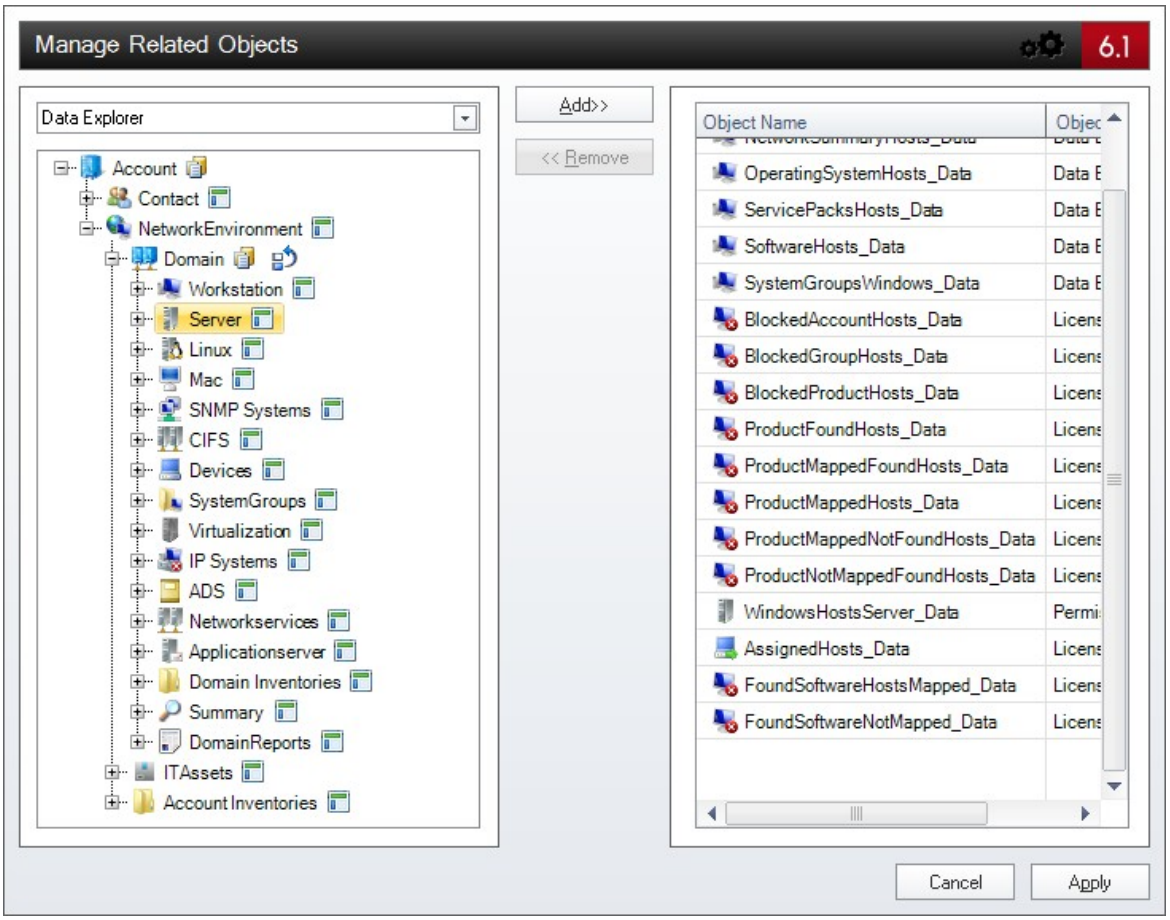


Related Objects

It is possible to use the same objects in multiple hierarchies. For example, the same computer can be displayed in the Data Explorer under the Workstations node and in the Permissions Analysis explorer. You can specify permissions and extensions for each computer. To make sure that this information will also be displayed for the same computer when listed in the Permissions Analysis hierarchy and do not have to be re-created, it is possible to define these two objects as related objects in the *Manage Metaobjects* dialog.

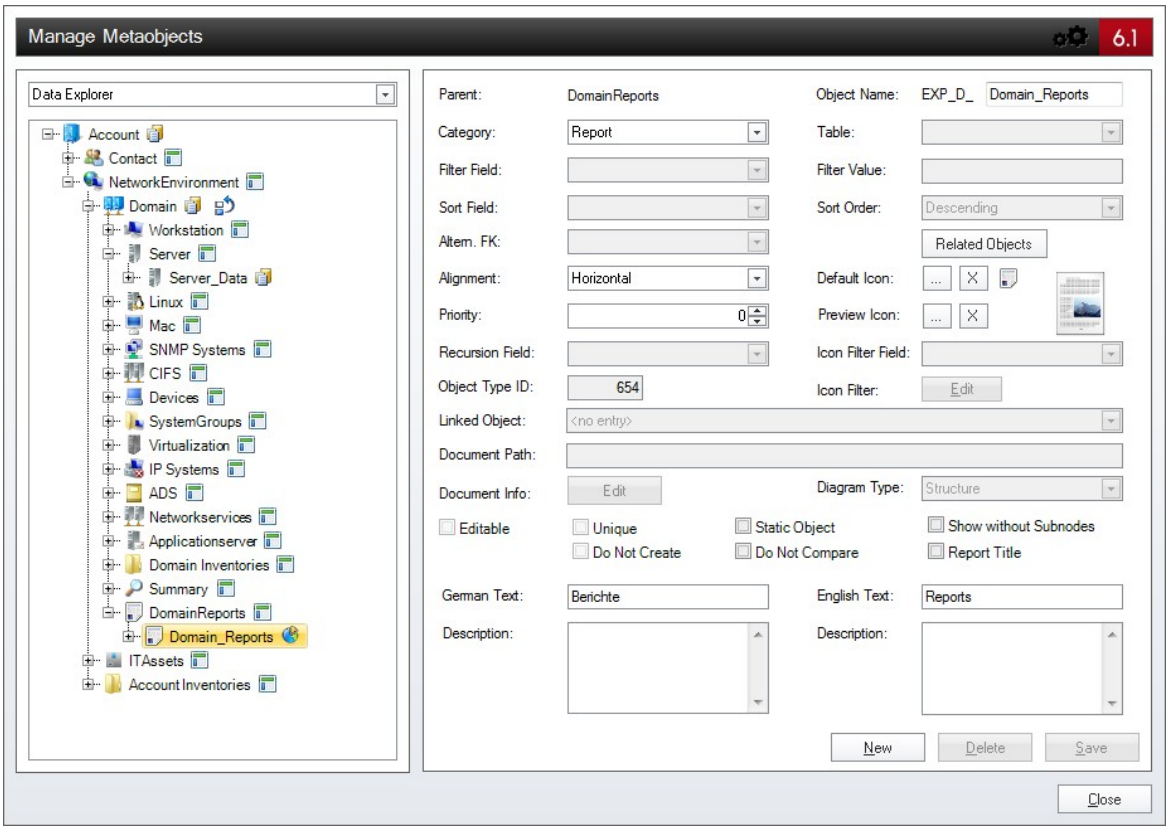
To do so, click the *Related Objects* button and select the related objects. In the left pane, all tree views will be displayed, and the right pane shows the objects that have already been defined as related types. When you select an object in the left pane, the *Add* button is enabled.

If you define related objects for the current object, it is recommended to repeat this assignment in the other direction, i.e. to assign this object to the related objects as well.



4.4 Reports

In the tree views, Report nodes serve as placeholders for the report definitions. The link between reports and meta objects is made from the *Manage Reports* dialog (Tools / Manage Reports). Any number of reports can be linked with a Report meta object.



For meta objects of the Report type, only the Alignment, Priority and Text properties are relevant

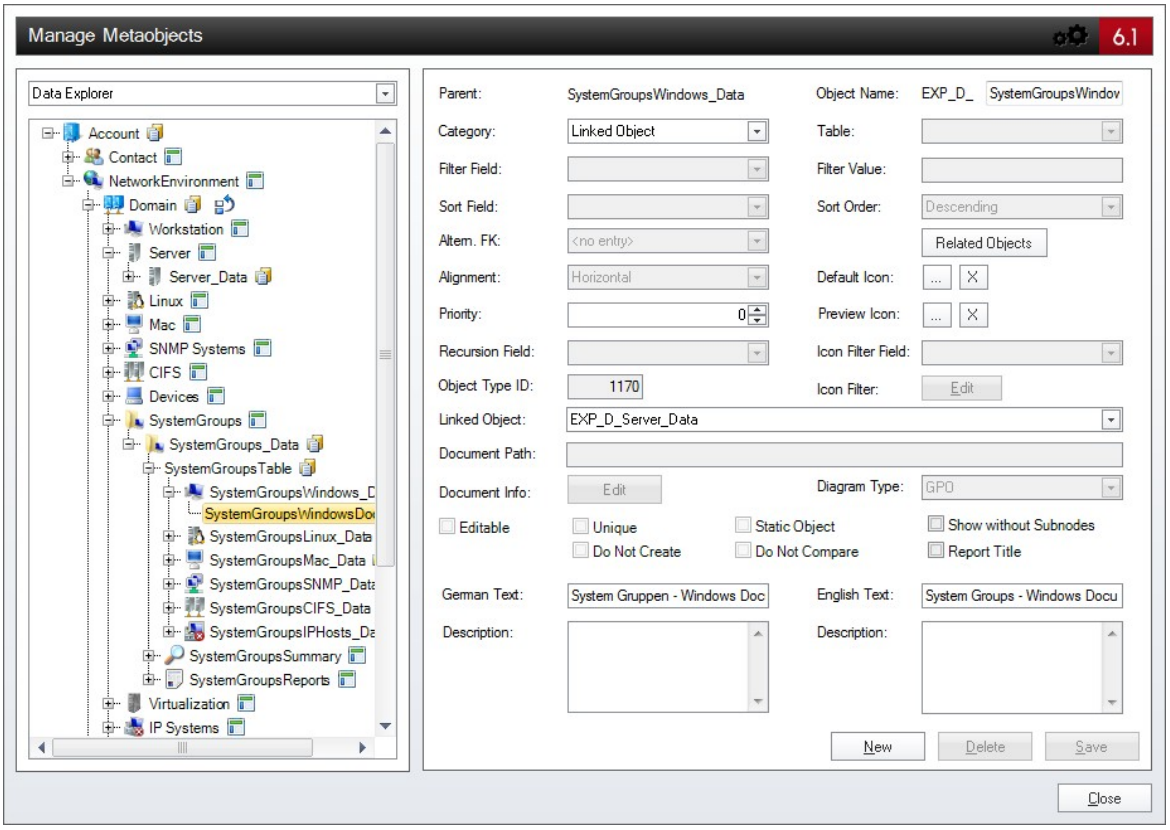
4.5 Links

Meta objects of the Linked Object type represent a logical link within the tree view. By creating a linked object, you can define cross-references in the structure and thereby reduce the number of objects to be defined.

For meta objects of the Linked Object type, you can select the target meta object from the *Linked Object* field.

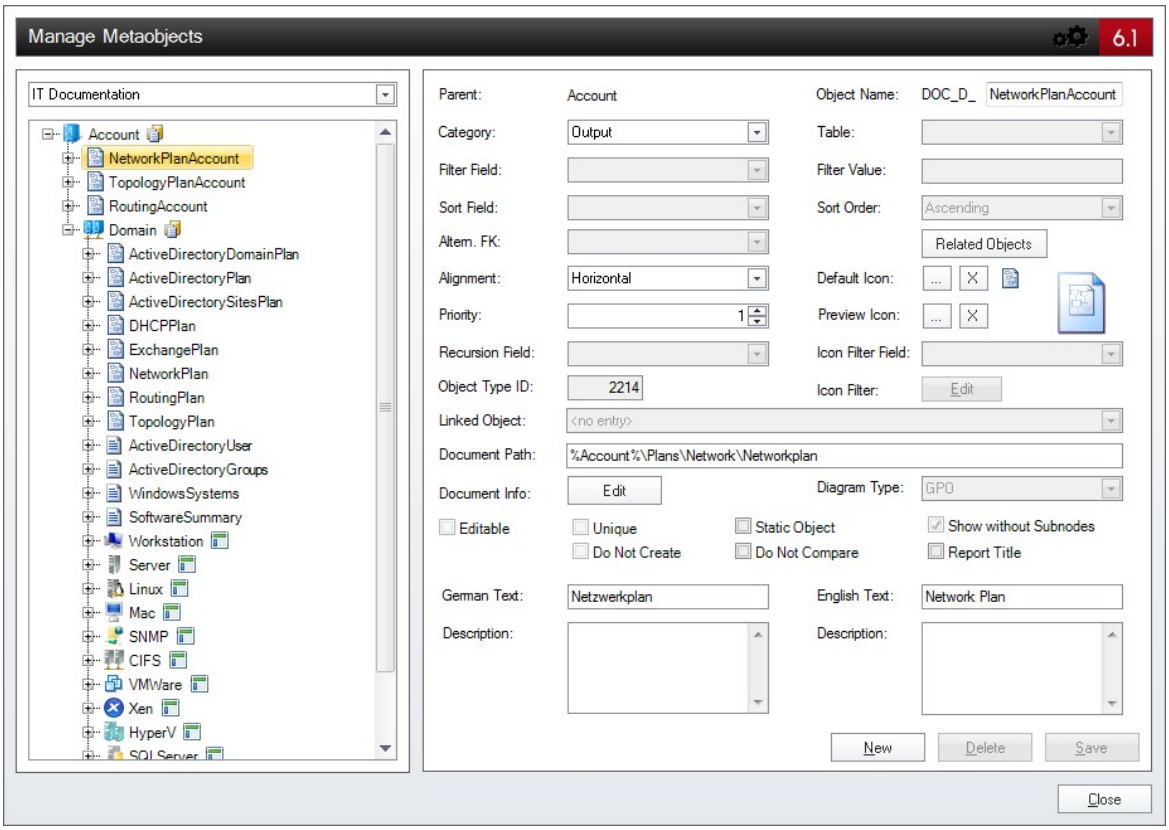
Linked objects are not taken into account for data comparisons and full-text searches because the actual data has already been compared or searched under its original path.

The target of the link must always match the primary key of the very meta object that is the parent of the Linked Object type meta object. Otherwise, Docusnap will not be able to select the data properly.



4.6 Output

Meta objects of the Output type are used to show documents such as plans, datasheets or overviews in the tree structure. Unlike the Document Path used for meta objects of the Data and Caption types, Docusnap will only display the *Document* tab for Output-type objects. In addition, you can specify which wizard will be used to create the document. The selected wizard will be shown as a dashboard on the *Document* tab if the corresponding document has not been created yet.

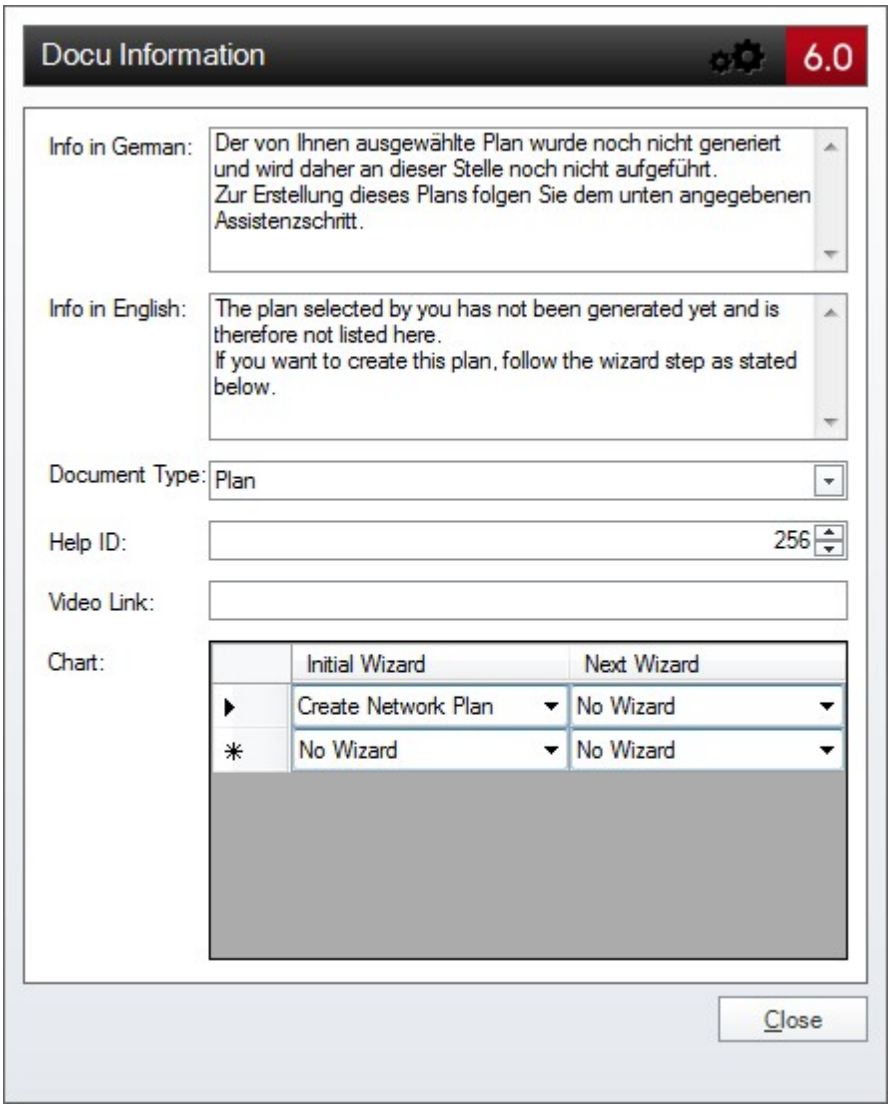


Property	Description
Document Path	The document path specifies the path to external documents, e.g. in Word or HTML format. When you select an object, Docusnap automatically checks whether documents exist in this path. If documents are found, they will be displayed on the Documents tab. The path entered for this property is always relative to the documentation directory. In order to make paths flexible, you can use variables when specifying the path. The following variables are available: %Account% Company name %Domain% Domain name %Object% Object name %ParentObject% Object name of the parent node

Property	Description
	Docusnap will automatically append "_EN.html" or "_DE.html" to the file names.

If the desired document has not been created yet, a dashboard displays. From there, you can launch the wizard for creating this document. Click the *Edit* button next to the *Document Info* field to open the *Docu Information* dialog.

Property	Description
Info in German / Info in English	In the *Info in German* and *Info in English* fields, you can enter the text to be displayed on the dashboard.
Document Type	In the *Document Type* field, you can specify whether the document is a plan, a datasheet or a list.
Help ID	Using the help ID, you can specify which section of the help system will be accessed via the dashboard.
Chart	In the *Chart* group, you can select the wizard to be used for creating the desired document. If another wizard is required after the initial wizard, you can select it from the *Next Wizard* dropdown list.



4.7 Diagram

Meta objects of the Diagram type are used to show the structures and Active Directory group policies as separate nodes in the Data Explorer. When defining a meta object of the Diagram type, you can select whether the diagram refers to a structure or a group policy object.

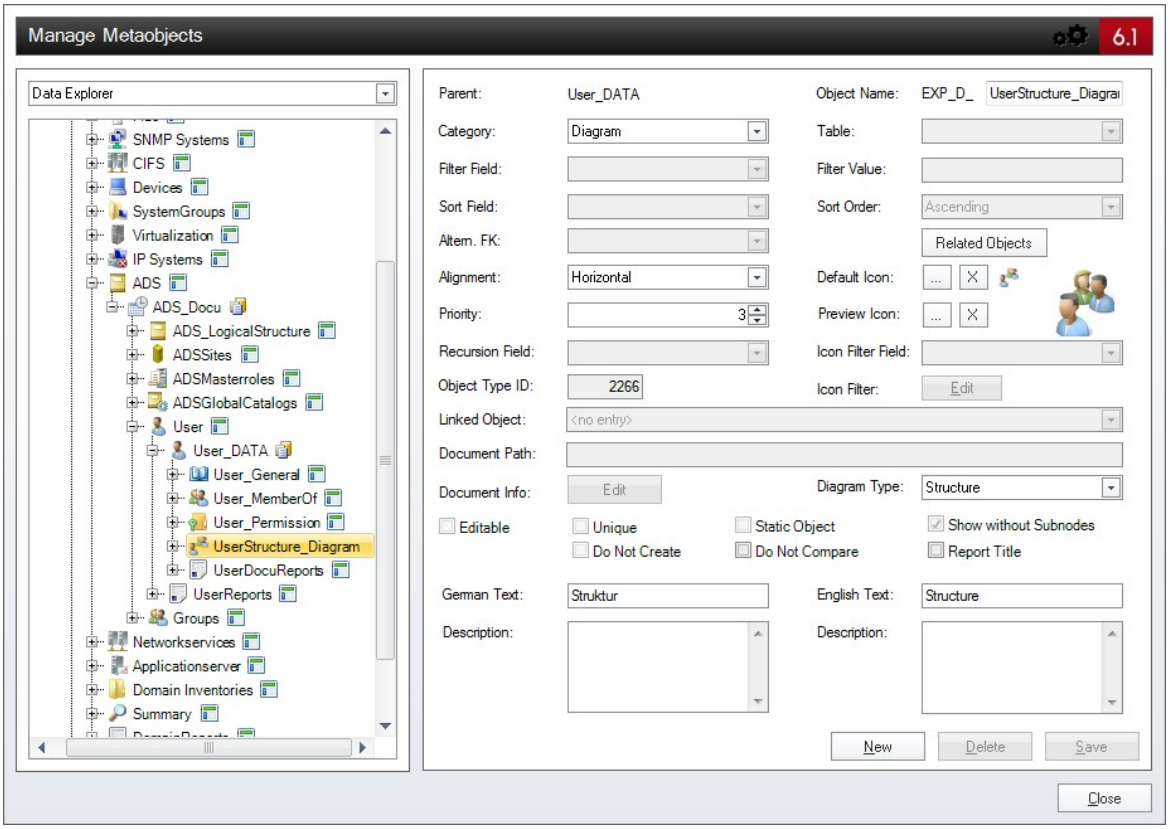
Structure

For Active Directory users and Active Directory groups, an additional Structure tab is used to show the structures. This tab shows group nesting as well as user nesting information.

Using this information, you can find out the following:

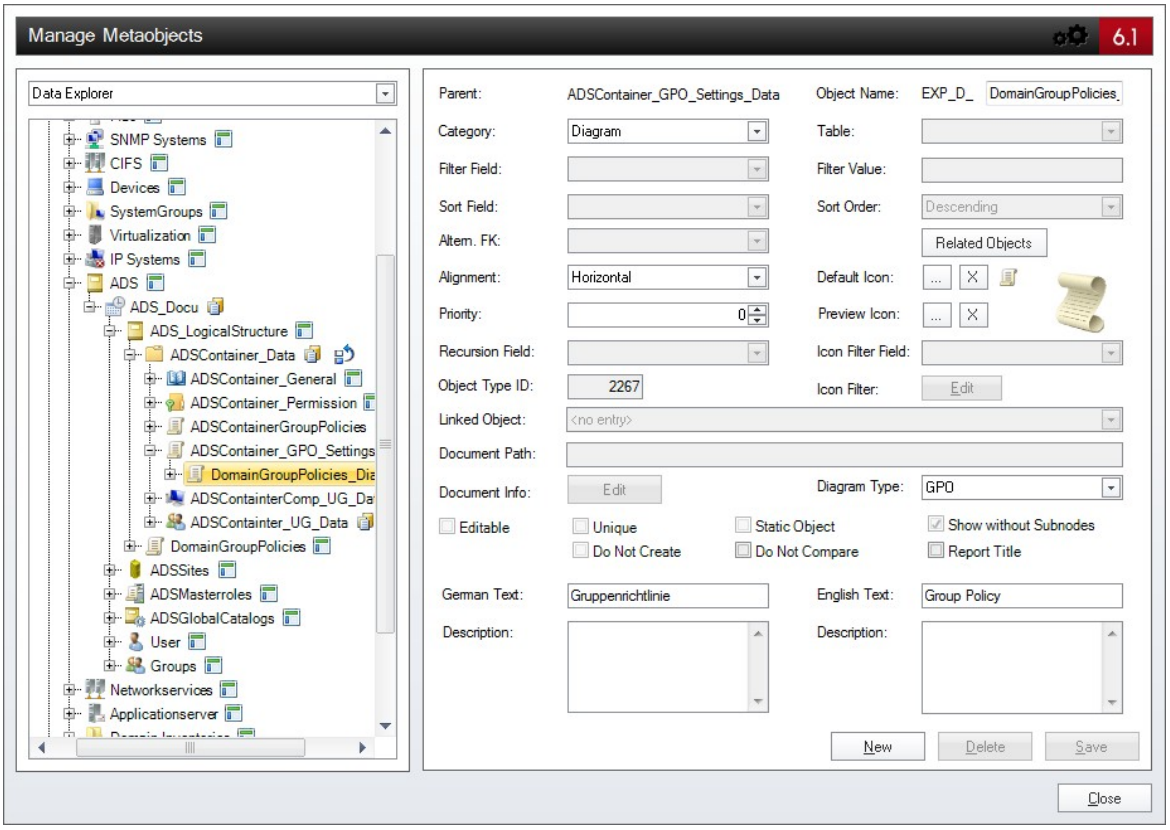
- Which group or groups is the user xxx a member of?
- Which members make up the group yyy?

The Diagram-type meta object can be used to show the structure as a separate node below the Active Directory user and Active Directory groups.



Group Policies

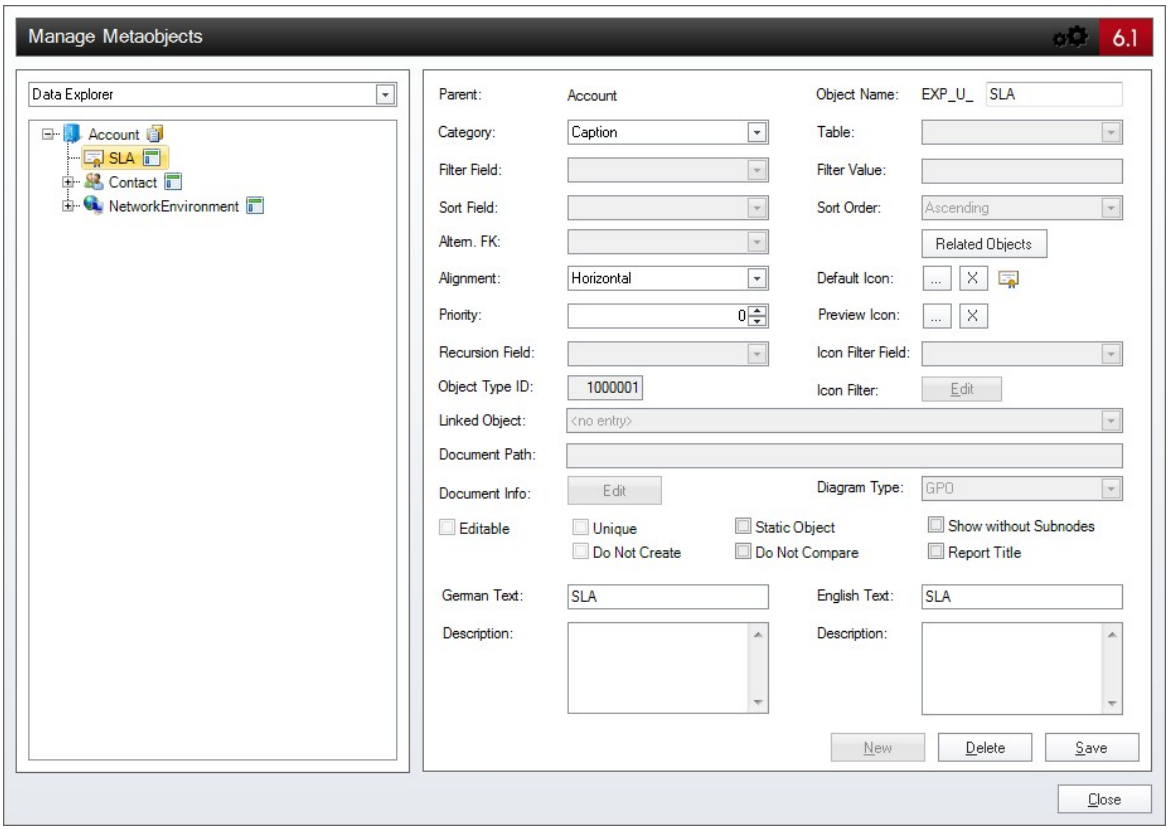
In Docusnap, you can retrieve group policies and assign them to the corresponding organizational units, sites and domains. When you create a meta object of the Diagram type and the GPO diagram type, Docusnap creates a group policy node that is displayed below the group policy information. As the table for the parent node, select one that uses the *GPOSettingsID* field of the *tADSGPOSettings* table as the primary key.



4.8 Example

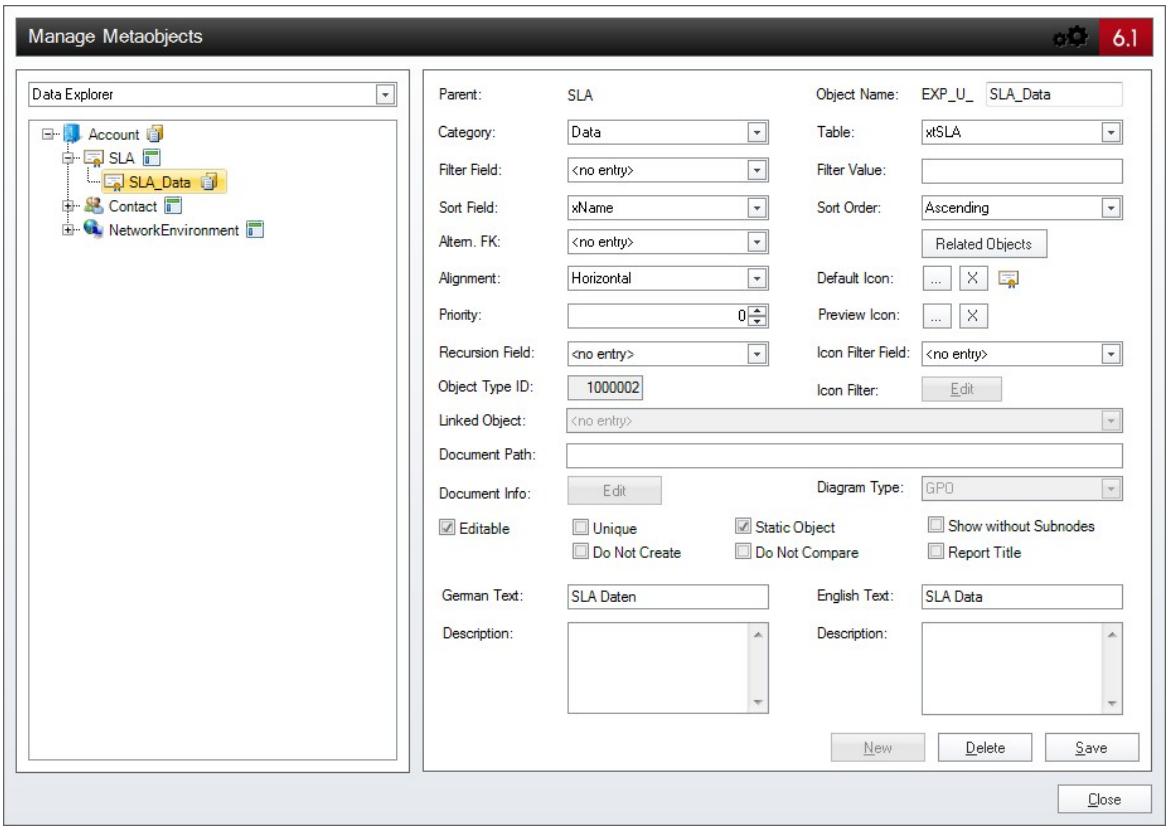
As a simple example for extending meta objects, we will demonstrate how to create the additional SLA object. To execute this example, first create the *SLA* table. For creating additional tables, refer to the Example subsection of the Database Structures section.

In the first step, create the *SLA* caption. Create a new object below the Account object in the Data Explorer by clicking the *New* button. Assign the *Caption* category to this object. Enter *SLA* as the Object Name. The English text and the German text are the same in this example. In addition, tick the *Static Object* and *Show Without Subnodes* checkboxes.



Save the caption and then create an additional object below it. Link this object with the *xtSLA* table. Select the *Data* category for this object and enter *SLA_Data* in the Object Name field. In the *Table* dropdown list, select the *xtSLA* table.

Enable the *Editable* checkbox so that it will be possible to enter data through the user interface. For information on how to create the data entry screen, refer to the Example subsection of the Data Entry Screens section. Save the object by clicking the *Save* button.



Part V

5 Data Entry Screens

Interaction with the Database

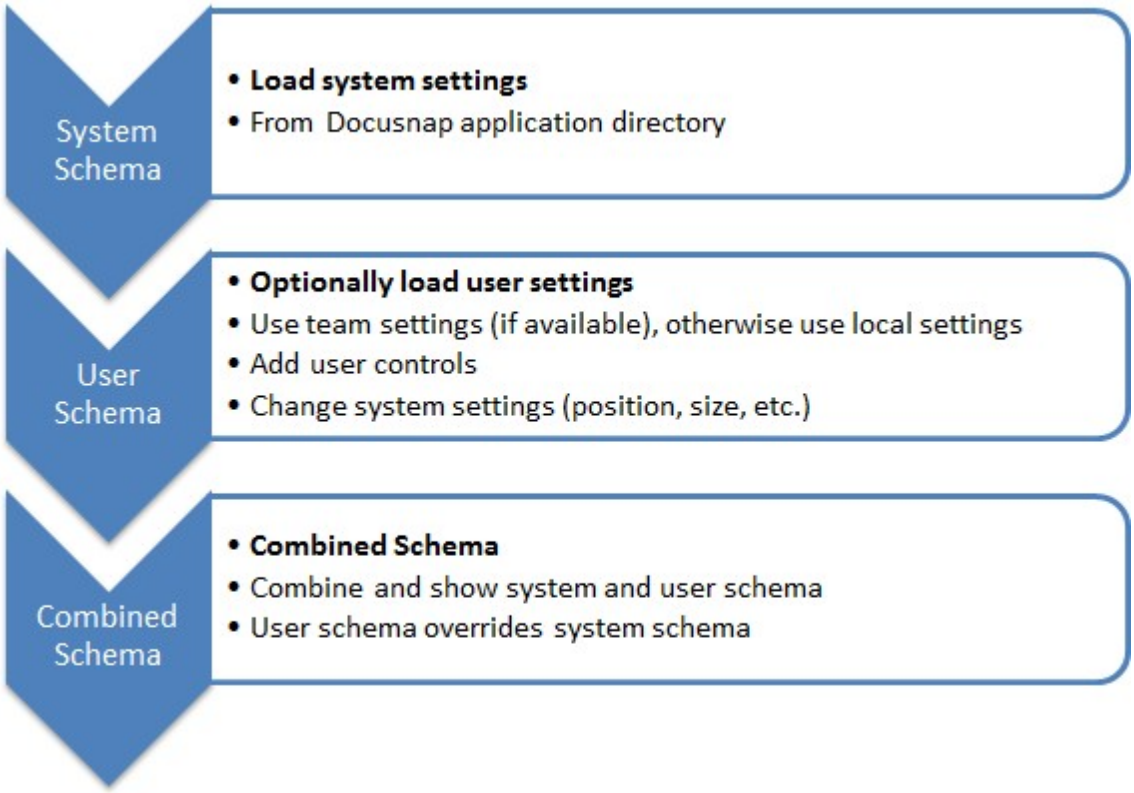
All editable meta objects have predefined data entry screens that allow fast and easy editing of all available data directly in the Docusnap tree view. Their function is that of an interface between the user and the database, performing a basic validation of the data entered by the user and generating error messages, if necessary. All system-defined data entry screens can be modified and extended by the user, so that maximum flexibility is achieved. In addition, you can create new data entry screens for user-defined meta objects from scratch. This way, and by extending the database and the object structure, it is possible to perfectly customize and extend Docusnap to meet all current corporate requirements.

System and User Schemas

The appearance of each data entry screen used in Docusnap is defined by a corresponding definition file stored in the program directory. In this context, please note that each predefined data entry screen has a so-called system schema with a .des file extension, which is located in the Dataedit subfolder of the Docusnap program directory. These files contain all system-defined values and enable you to reset modified data entry screens to their original configuration. In addition to these files, edits and extensions made by the user are stored in so-called "user schema" files with a .deu file extension. Depending upon the relevant setting, these files are stored in the *DataEdit* subfolder of either the local or the team settings folders for Docusnap.

Docusnap always loads the system settings for each data entry screen first and, if user settings exist, uses them to customize or extend the data entry screen. It should however be noted that not all of the predefined settings can be replaced by user settings. For example, it is not possible to delete predefined controls or to change the links to the database. But you can change the size and position of each control as required. There are no such limitations with regard to user-defined extensions. All controls created by users can be deleted as desired. The illustration below shows how the user and system schemas are used in Docusnap.

Data Entry Screens

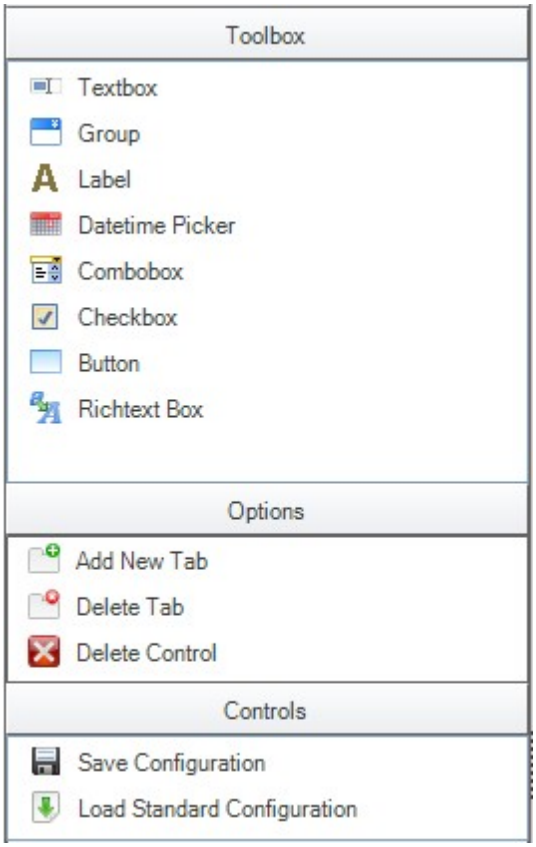


Due to the fact that data entry screens primarily represent the interfaces to the current Docusnap database, it should be noted that controls that can be used to enter or manipulate data will automatically be disabled if their configuration is invalid. This might be the case if no link or an invalid link to the database has been specified for a control.

After changes have been made to any data entry screens, you might need to restart Docusnap or close the entire tree structure and re-open it in order to load the changes.

5.1 Designer

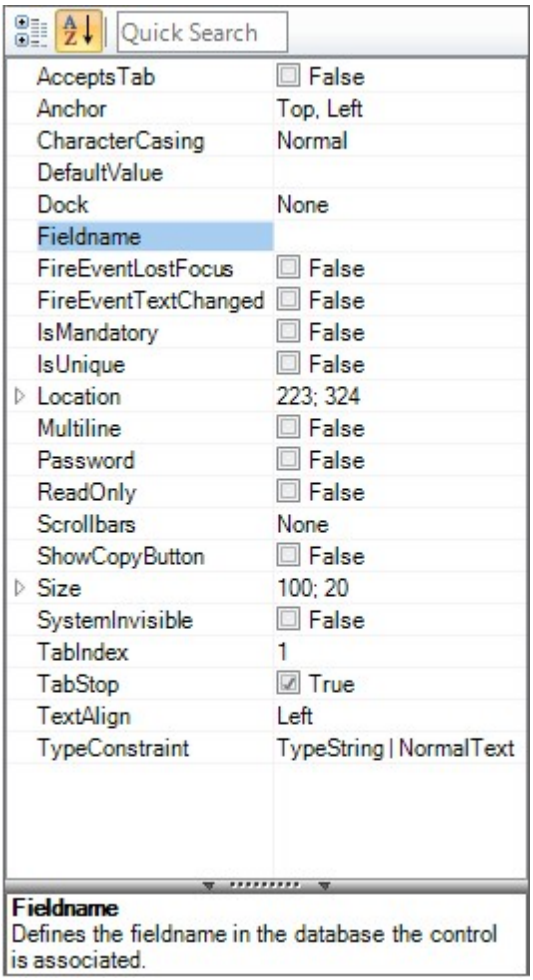
Basic Structure of the Designer



In Docusnap, an integrated Designer is available that you can use to customize or create data entry screens. In can be opened by clicking the Design Mode button on the Tools ribbon. To start the Designer, an editable meta object must have already been selected in the Data Explorer. As long as the Designer is active, all input and output processes on the data entry screens are disabled and the controls are set to an editing mode. In order to close the Designer and continue working with the actual Docusnap user interface, simply click the Design Mode button again.

Basically, the Designer for editing data entry screens consists of three panes: the Workspace, the Toolbox and the Properties tab. The Workspace is the largest of these areas. It is located in the right Designer pane and has an orange border. Above the workspace, the so-called "tabs" are located. They extend the available space and help to organize the data entry screens in a meaningful way. They will be explained in more detail in the Using Tabs section. In the left Designer pane, either the Toolbox or the Properties tab is displayed. To save the edits made to a data entry screen, click the Save Configuration button in the Toolbox (you may need to scroll down the pane if it is not visible). Similarly, you can reset a data entry screen to its original settings by clicking the Load Standard Configuration button.

Adding and Configuring Controls



To add a new control, simply drag it from the Toolbox and drop it on the Workspace. At the top of the Toolbox, various controls are available that can be used in different situations. For details regarding the functionality and properties of the each control, see the Controls section. There are two ways to remove an existing control from a data entry screen. After the control has been selected (is is highlighted by a red border), you can delete it by either clicking the *Delete Control* button in the Toolbox or by pressing the *<Del>* key on the keyboard.

To move a previously added control, simply select it and drag it to the desired position with the mouse. Precise alignment can be done using the X & Y coordinates on the Properties tab. These properties, like any other available properties, can be changed from the Properties tab. Please note that the properties of system-defined controls may differ from those of controls added by a user. Similarly, predefined controls cannot be deleted. To modify the properties of a control by means of the Properties tab, the desired value can, in most cases, be entered directly into the right column of the displayed table. In some special cases, it is possible to make changes to the settings from a separate dialog. If so, a button with an ellipsis (three dots) is displayed in the right column of the Properties tab when you select such a setting. When you click this button, the dialog opens where you can specify the corresponding settings.

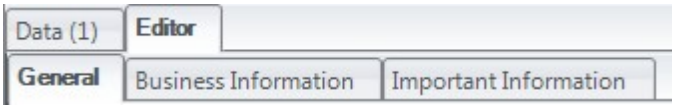
All controls that are placed on a data entry screen will automatically be linked with the current record for the associated meta object and can be used to edit this data.

5.2 Usage of Tabs

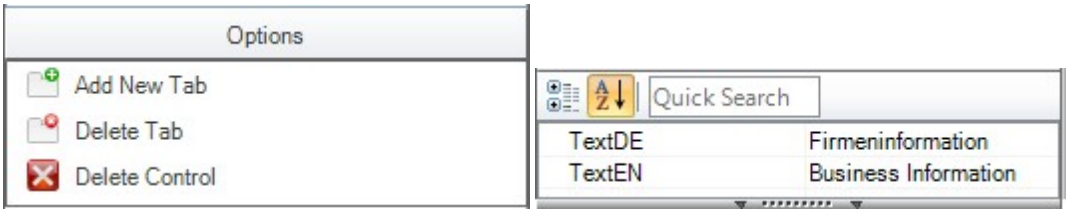
Usually, a data entry screen consists of several so-called "tabs". Various controls can be placed on each of these tabs, all of them being related to the same record in

the current Docusnap database. Each of the predefined data entry screens has a tab named *General*. In almost every case, it contains the predefined controls for Docusnap.

Since the available space is thereby already quite limited in many cases, it is possible to add any number of additional tabs to the data entry screen. This way, the available space can be extended as desired. At runtime, you can switch back and forth between the individual tabs as desired.



In the *Options* section of the Toolbox in the left Designer pane, buttons for the creation and deletion of tabs are available. To specify the name of a tab, select the tab in the Designer, open the Properties tab and enter the desired text for the *TextDE* and *TextEN* properties.



5.3 Controls

Available Controls

The following eight controls are available to customize and extend the Docusnap data entry screens. Their properties will be described in the following sections:

- Textbox
- Group
- Label
- Date/Time Picker
- Combo Box
- Checkbox
- Button
- Rich Text Box
- Tree View

General Properties

In addition to the specific properties of the individual controls, several general properties are available, which serve a similar purpose for most of the available controls. The two tables below briefly list these two groups of properties, including an explanation.

Global Properties of All Controls	
Anchor	Determines the edges where the current control is anchored to a parent control. For data entry screens, the parent control can be either a group, in which the respective control is located, or the data entry screen itself. The default setting for the anchor is *Top, Left*. This means that the the upper left corner will be anchored, which, at runtime, results in a constant offset between the control and the upper left corner of its parent control. If, for example, you want the size of a control to grow or shrink horizontally with an increasing or decreasing resolution, then a *Left, Right* anchor would be necessary. If you further want the vertical position to remain constant, it would be recommended to use an additional *Top* anchor. If vertical size adjustment is desired as well, you can also set the *Bottom* anchor. You can easily select or deselect an anchoring option by clicking the down arrow of the *Anchor* property on the *Properties* tab and then selecting the corresponding area.
Dock	This property defines whether the current control will be aligned with the the parent control and, if yes, at which edges. For this property, the behavior of the control is the same as described for the *Anchor* property. When you select a docking edge for the control, it will always be attached to this edge of the parent control, regardless of its own size and other properties. If you select the *Fill*docking option, the control will fill the entire surface area of its parent control. This makes sense, for example, if you want a text box to use the entire available area. If you rather want to leave a small margin between the text box and the parent control, it would be better to use the *Anchor* property instead and select all four options.
Location	This property specifies the position of the control relative to the upper left corner of the parent control. The values are based on a Cartesian coordinate system with the origin in the upper left corner.

Size	Defines the size of the control. For the values, the "width x height" pattern applies. Please note that size limitations exist for certain controls. For example, the height of a single-line text box cannot be increased. In addition, other properties, such as *Dock* or *Anchor*, may change the size of the control.
TabIndex	This property defines the sequence in which the controls on a data entry screen will be selected when a user presses the <Tab> key. Enter an integer for this property. The controls will be accessed in the order defined by the numbers entered for each control in this property, starting with the smallest.
TabStop	This property must be set to allow a user to select the corresponding control by pressing the <Tab> key. Otherwise, the control will be skipped, when a user presses the <Tab> key and the next control where this property is set to 'True' will be accessed.

Properties for Some Controls	
DefaultValue	This is the default value of the control. The control will be set to this value when the control is first initialized. It can be overwritten by the user, if required.
IsMandatory	Indicates if a value must be specified for the control. If this property is set to 'True' and the user did not enter or select a value, the data cannot be saved and an error message displays. It is a good idea to set this property for all fields where the *NOT NULL* flag is set in the database schema. But this property can also be applied to fields that may be set to *NULL*.
IsUnique	If this property is set to 'True', Docusnap validates the data the user entered or specified for this control when saving. If duplicates are found, the data cannot be saved and an error message displays. This validation is done at the table level. This way, Docusnap makes sure that no identical record exists in the database table.
TextDE/TextEN	Using these two properties, you can specify an English and a German caption for this control. *TextDE* represents the German

	caption and *TextEN* the English caption. For most controls where this property is available, this is the text that will be displayed on the user interface.

5.3.1 Field

Purpose and Properties

In most situations, text boxes are the most important controls of data entry screens. Primarily, they are used to enter free text, but you can assign comprehensive formatting and validation options, if required. This allows you to use text boxes to prepare data entry screens for the input of the most diverse data. A special case is the so-called number servers which automatically populates a text box, based on a predefined schema.

The table below list the specific properties of text boxes, including a brief explanation:

Specific Properties of Text Box Controls	
AcceptsTab	Specifies whether the tab key can be used as an input character or not.
CharacterCasing	Defines whether the text entered by a user will be shown in uppercase, lowercase or normal characters.
Fieldname	Specifies the linked column in the respective table of the current Docusnap database. The meta object linked with the data entry screen determines which table will be used. Unlike most other controls, text boxes can be linked with nearly every data type that exists in the database. Make sure to specify an appropriate data validation, otherwise errors may arise when users enter data of the wrong type.
Multiline	Determines whether a text box should consist of one single line or multiple lines. It is neither possible to enter line breaks into single-line text boxes nor can the height of these text boxes be changed. Multi-line text boxes are recommended for the input of large amounts of information. Care should be taken to ensure that the corresponding database field is large enough to accommodate the data (use

	the varchar(1500), TEXT or MEMO type).
Password	If you set this property to 'True', each character entered by a user will be displayed as an asterisk (*). In addition, two buttons will appear next to the text box. When a user clicks the first one, the content of the text box is displayed as plain text. The other one copies the current content of the text box to the clipboard from where it can then be pasted into other applications.
ReadOnly	Using this property, you can specify whether the text box will be used for data input and output or merely for the display of database data. If you set this property to 'True', users will not be able to change the data in the linked database column. This property is always useful if the data shown in the text box are predefined and should not be changed. However, please note that there must always be another way to save the data in the database, for example, by means of a script or an inventory process. One use case for this property would be when using the number server, because the content of the text box is generated automatically and users should be prevented from changing it.
Scrollbars	Determines if scroll bars will be displayed in (multi-line) text boxes and if so, which type. In principle, navigation within a text box control is also possible without scroll bars by using the keyboard or the mouse. However, scroll bars significantly facilitate this process.
ShowCopyButton	Specifies if a button will be displayed in the right part of the text box for copying its current content to the clipboard.
SystemInvisible	If you set this property to 'True', the text box will not be displayed during the runtime of the data entry screen, but rather filled with a predefined value that you can determine by means of the*DefaultValue* property. This approach is useful when the database schema requires a static value that the user cannot enter, for example automatic input of the corresponding device type in data entry screens at the device level.

TextAlign	Determines how the characters will be aligned in the text box. As a rule of thumb, select left alignment for free text and right alignment for numbers (this improves their readability).
TypeConstraint	Defines specific validation options for the content of a text box. For details on these options, see the subsection below and the dedicated number servers section.

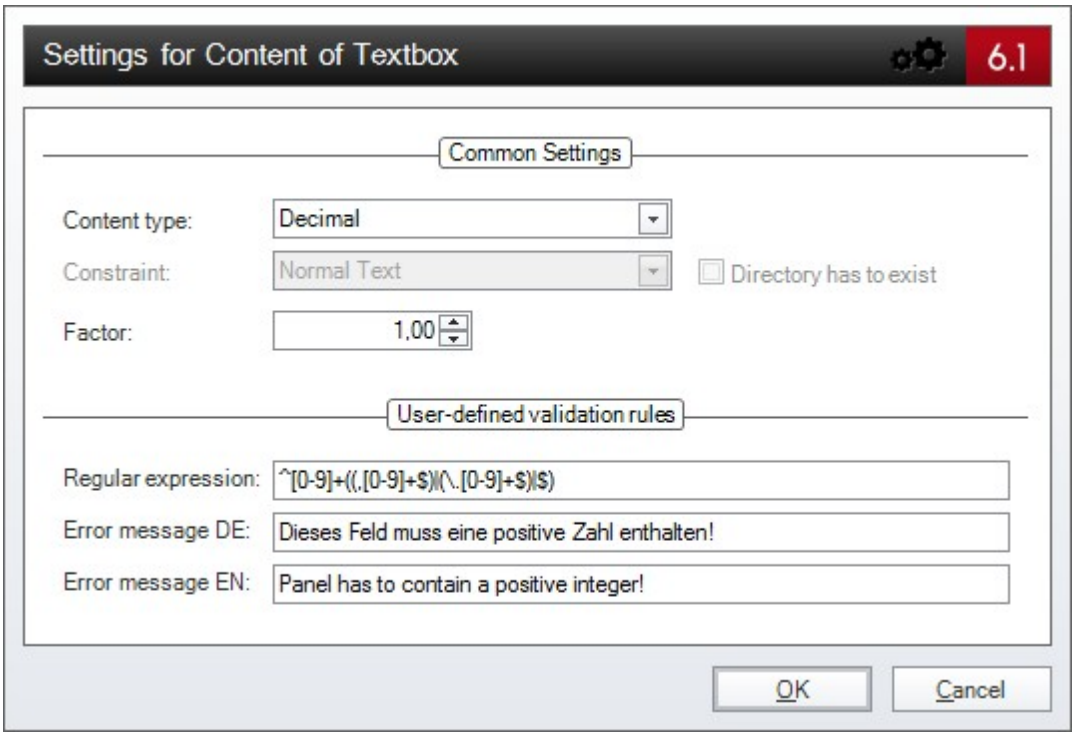
TypeConstraint Property

This property can be used to store specific validation options for the input of data in a certain text box. If the validation results in errors when a user tries to save a record of data entered into the respective data entry screen, the save will be aborted and a predefined error message will display. Several predefined validation types are available that are briefly explained in the table below.

Content Type	Description
Text	When you select the *Text* content type, users can enter free text into the text box that might be checked further before they are saved. • ***Normal Text***: Users can enter any text. It will not be validated. • ***IP Address***: Verifies if the text entered by the user is a valid IP address. • ***Subnet Mask***: Verifies if the text entered by the user is a valid subnet mask. • ***MAC Address***: Verifies if the text entered by the user is a valid MAC address. The character groups may be separated either by no character at all or by colons (:) or hyphens (-). • ***Directory***: Verifies if the text entered by the user is a valid directory path. If you also enable the *Directory has to exist* checkbox, Docusnap will check whether the specified directory really exists on the hard disk or share. • ***Drive Letter***: Verifies if the user entered a valid drive letter. Please note that Docusnap will not check whether the drive really exists and is connected..

Integer	Specifies that this text box is reserved for entering integers. For number values, you can also specify a factor in the Factor field so that the integer entered by the user will be multiplied or divided by this value. The database will store the product of this multiplication. In this case, it is recommended to create a corresponding column for the factor in the meta table to obtain a consistent representation of the numbers in the data entry screen and the associated lists.
Date	Specifies that this text box is reserved for entering dates.
Byte	Specifies that this text box is reserved for entering byte values.
Memo	Specifies that this text box is reserved for entering text in the MEMO format.
Binaray Data	Specifies that this text box is reserved for entering binary data.
Bool	Specifies that this text box is reserved for entering Boolean data values.
Decimal	Specifies that this text box is reserved for entering decimal numbers. As with integers, you can specify a factor.

In addition to the predefined types, you can define and apply your own validation rules. When defining such rules in Docusnap, you can use regular expressions (RegEx), which allow the definition of almost any validation check. Introductory information about regular expressions can be found on many websites, such as Microsoft's EN http://msdn.microsoft.com/en-en/library/az24scfc.aspx or Wikipedia http:// en.wikipedia.org/wiki/Regex. To test and optimize regular expressions, we recommend the *Expresso* software from *Ultrapico* available under http://www.ultrapico.com/Expresso.htm.



Number servers are a special case of text box. For more details, see the Number Server section.

Examples of Regular Expressions

Use regular expressions to validate the input in a text box. The text entered by a user is compared with a predefined pattern to make sure that the text is, for example, an e-mail address, a positive number, etc.

What do regular expressions consist of?

Enclose valid letters, digits and characters in square brackets []. The following characters can be used to define how often these letters, digits and characters should be repeated: ?,+ and *.

Characters	Meaning
[A-Za-z]	This expression checks if a letter of the Latin alphabet has been entered at the indicated position.
[0-9]	This expression is used to find any digit at a certain position in the input.
?	The preceding letters, digits or characters are optional. They may (but need not) occur once, i.e. the expression occurs once or not at all.
+	The preceding letters, digits or characters must appear at least once,

	but may occur repeatedly.
*	The preceding letters, digits or characters may be repeated any number of times (or may not occur at all).
|	The | symbol can be used to define a logical OR. Place the | symbol between the alternatives.
^	Use the ^ character to identify the beginning of the pattern.
$	The $ symbol is the counterpart of the ^. It indicates the end of the pattern that has to match the end of the string. In a regular expression, you can define multiple end characters, provided that the expressions are separated by a logical Or (|).
.	Enter a period to check that any character (with the exception of a line break) exists at a certain position.
-	Within character classes, a hyphen is interpreted as a character. There, it is used to specify character ranges.
\	If you want to use a character in the pattern that has a specific meaning, such the period as a punctuation mark and not in its meaning as an expression for a given character, precede it with a backslash (\) to "escape" it.
()	To group expressions, use parentheses.

Examples:

Check if a number is positive:

```
^[0-9]+((.[0-9]+$|(\.[0-9]+$)|$)
```

Check if the user input represents an e-mail address:

```
^([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})$
```

5.3.2 Groups

Group controls can be used to organize other controls, in order to design more clearly structured data entry screens. From a functional point of view, controls in groups do not make any difference to controls that have been placed directly on the data entry screen. Controls organized in groups also use the database record of the parent control (in this case, always the data entry screen itself). However, you can move child controls together with their group control, and properties like *Anchor* or *Dock* use the coordinates of the group control for their origin.

Like all other controls, you can position a group control simply by dragging and dropping it on the data entry screen. To place a control within in a group, drag it

from the Toolbox directly onto the group control. It is neither possible to insert an existing control into a group control at a later time nor to move a group control into another group control or into the parent control. Besides, group controls cannot be nested within other group controls. This means that only one hierarchical level is allowed.

In addition to the global properties, a group control only has one additional property named *Border*. This property controls the visual appearance of the group control in the data entry screen. Two options are available: By selecting the *AllSides* option, you can display a border around the entire group control. In contrast, the *OnlyTop* option displays only a single line along the upper edge, while the other three sides are left open.

5.3.3 Caption

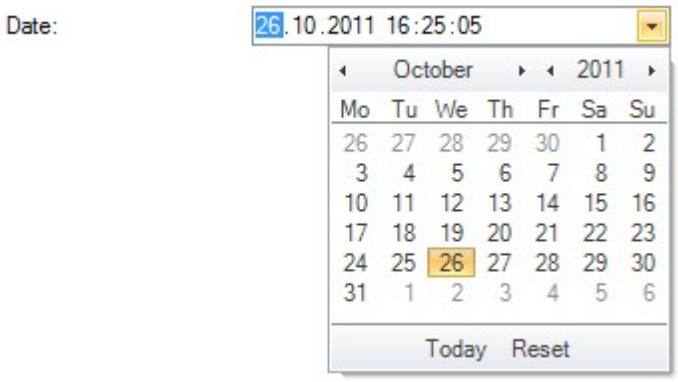
Like textboxes, labels can be used for the output of text. However, the user cannot enter data into labels, as they are merely used for display purposes. Usually, labels describe or name other controls on the data entry screen. Thus, a label can indicate the purpose of a text box.

In addition to the global properties, labels have two additional properties, which are explained in the table below.

Specific Properties of Labels	
AutoEllipsis	If this property is set to 'True' and the text exceeds the size of the control, an ellipsis symbol (...) appears at the right edge of the control to indicate that the text continues. If this property is set to 'False', the text will simply be truncated at the edge of the control..
TextAlign	Similar to the property of the same name for text boxes. This property determines the horizontal alignment of the string it contains.

5.3.4 Date/Time Picker

By adding a *date/time picker* to your data entry screen, you can provide a control that helps the user select a time or date to be saved in the database. This control is useful if users have to enter periods of validity, expiration dates and similar values. Optionally, a date selection may of course be made using a text box, provided that you have configured it with the corresponding type and validation settings.



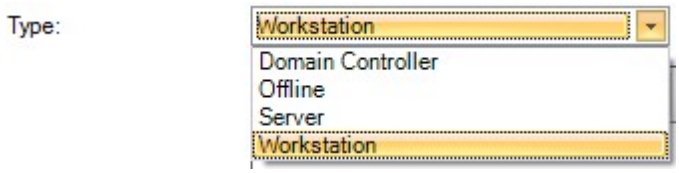
Specific Properties of Date/Time Pickers	
DisplayFormat	This property defines how the date will be represented in the control. Five different options are available. How the date or time will be displayed at runtime depends upon the locale settings in the control panel of the computer. The following options are available: • ShortDate: date in short format • LongDate: date in long format • Time: time in long format • DateTime: date in short format, time in long format • DateTimeShort: date and time in short format
Fieldname	Specifies the linked column in the respective table of the current Docusnap database. The meta object linked with the data entry screen determines which table will be used. For a *date/time picker* control, the corresponding target column in the database must always be of the Date data type.
MaxDate	Specifies the latest date the user will be able to enter or select for this control

MinDate	Specifies the earliest date the user will be able to enter or select for this control.

5.3.5 Combobox

So-called "combo boxes" display a list of valid values the user can select from. These controls are also known as drop-down lists. For this purpose, all valid values for the linked database column are presented to the user in a list from which the desired value can then be selected. This control is therefore suitable in all cases where a limited number of valid values are available that may change dynamically. As the data source, other database tables or Docusnap reference values may be used. For a list of all reference values including explanations, see the appendix. If you select the database as the data source, you can specify any existing database table, provided that a primary key and a display field have been defined for it. When a user selects a value from this control, Docusnap will not save the selected string, but rather its associated number, which corresponds to the primary key (if the data source is a table) or the respective value (if the data source is a reference value).

In many cases, it is desired that only certain values are available from the combo box, such as only contacts stored for the current company. To make this possible, you can use a view that is filtered using the {FilterID} variable. The {FilterID} will always be replaced with the primary key of the parent object.



The table below lists all specific properties for *combo boxes*.

Specific Properties of Combo Box Controls	
DropDownHeight	Determines the size of the dropdown list that should be displayed for the selection of the predefined values.
Fieldname	Specifies the linked column in the respective table of the current Docusnap database. The meta object linked with the data entry screen determines which table will be used. For a combo box, the corresponding target column in the database must always be a number that will be replaced with the associated text at runtime..
Sorted	If you set this property to 'True', the content of the combo

	box will be displayed in alphabetical order. Please note that reference values cannot be sorted explicitly. Using this option when *SourceType* is set to 'DCInitials' can lead to an erroneous behavior when the user selects a value.
SourceType	This property specifies the type of data source to be used for the combo box. As explained above, you can either use other database tables or the Docusnap reference values which are explained in more detail in the appendix of this manual.
SourceValue	Depending on the selected *SourceType*, enter either the name of the corresponding database table or the identifier of the desired reference value for this property.
NoSelection	Is this property is set to 'True', the combo box will also display a *<No Selection>* item. If the user selects *<No Selection>*, no value will be saved in the database. If the data column must be filled, this property cannot be used.

5.3.6 Checkbox

If the user needs to enter a Boolean value into the database from a data entry screen, a checkbox is helpful. Here, the checked control represents the "true" or "yes" value, and the unchecked control the "false" or "no" value. Checkboxes may either be used as independent controls or as label fields that have an additional feature used to enable or disable the control.

In addition to the global properties, checkboxes have the properties shown in the table below.

Specific Properties of Checkbox Controls	
CheckBoxPosition	This property defines where the actual checkbox should appear within the control. Docusnap uses its caption as the reference position.
Fieldname	Specifies the linked column in the respective table of the current Docusnap database. The meta object linked with the data entry screen determines which table will be used. For checkboxes, the corresponding target column in the

	database must always have the Boolean data type (BOOLEAN, YES/NO).

5.3.7 Buttons

Use and Properties

Buttons can be used to implement additional user-defined features directly on Docusnap data entry screens. For this purpose, either external applications with corresponding parameters may be started or VB scripts may be executed. In both cases, the data from the current record will be available and may be integrated into the corresponding program call or the script. To load the desired data, enter the corresponding database column names in braces in the following format: {FIELDNAME}

In addition to the global properties, the following property settings are available for buttons.

Specific Properties of Button Controls	
ActionMode	This property determines the type of action to be executed. The available options are *Application* or *Script*. When you select *Application*, clicking the button will start the application specified in the *Application* property. Additional parameters can be indicated by means of the *AppArguments* property. When you select *Script*, clicking the button will execute a VB script that you define through the *Script* property. The unselected option will automatically be ignored by Docusnap. Settings you have made for the currently disabled option will not be considered.
Application	Here, you can specify the application to be executed when the button is clicked. This requires that the *Application* option has been selected for the *ActionMode* property. Enter either the name of the application, such as *explorer.exe*, or the full path to an executable file on the hard disk or on a share.
AppArguments	Using this property, you can define additional arguments that will be passed as parameters to the application to be

	executed when the user clicks the button control. As with the Windows command line, multiple parameters can be entered. It is also possible to use data from the current record by entering the respective field name in braces.
Script	In this property, you can specify a VB script to be executed when the user clicks the button. Here again, data from the current record is available without restriction. You can integrate it into the script by entering the respective field name in braces.

Examples of Use

Opening an Admin Share

Specify the following to enable a user to open the C$ admin share of a Windows computer by clicking a button on a Windows systems level data entry screen:

Property	Value
ActionMode	Application
Application	Explorer.exe
AppArguments	\\{Hostname}\C$

Starting a Remote Desktop Connection

The configuration below enables the user to start a remote desktop connection for the current computer from a data entry screen at the device level.

Property	Value
ActionMode	Application
Application	mstsc.exe
AppArguments	/v {Hostname}

Simple VB Script Example

The simple example of a device level script shown below checks whether the currently selected computer is online or unreachable using a ping command.

Property	Value
ActionMode	Script

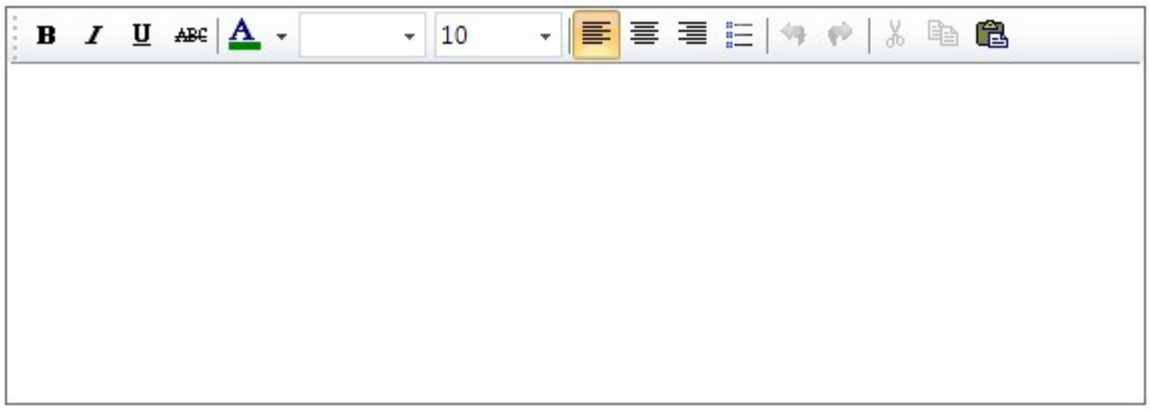
Script	Dim Win
	Win="select*from Win32_PingStatus where address='{Hostname}'"
	Dim Level
	Level = "winmgmts:{impersonationLevel=impersonate}"
	
	Set objPing = GetObject(Level).ExecQuery(Win)
	For Each objStatus in objPing
	If IsNull(objStatus.StatusCode) Or objStatus.StatusCode<>0 Then
	WScript.Echo "Computer {Hostname} is unreachable."
	Else
	WScript.Echo "Computer {Hostname} is online."
	End If
	Next

The examples shown in this section are meant to provide a basic look at the options to extend Docusnap by integrating external applications or VB scripts. Almost any imaginable customization can be made by means of these two options. If a planned customization goes beyond the capabilities of a VB script, you can call a self-programmed application with corresponding parameters in order to achieve the desired goal. More information about the capabilities provided by VB scripts can be found in a number of books and, of course, on the Internet.

5.3.8 Richtextbox

Rich text boxes enable the user to enter and edit formatted text at runtime. Thus, the user can benefit from all format options that have been provided by Microsoft for the corresponding text entry fields. Similar to a mini-word processor, users can apply simple formatting to the text as well as paste graphics, tables, photos and other objects simply by using copy & paste. Elements copied from another applications (or other parts of Docusnap) can be pasted at the current cursor position by pressing the *<Ctrl>+<V>* hotkey or by clicking the corresponding button on the toolbar, provided that this functionality is supported by the rich text box. Thus, rich text boxes are particularly suitable to enter comprehensive passages of text such as notes, comments, or they can simply be used as a means to enter formatted text.

When using rich text boxes, please note that the formatting applied to the text will be saved in the database as strings. When creating database fields that save formatted text, make sure that they are large enough for this purpose. Example: A length of varchar(255) will be sufficient in only a few cases, because this character count will be easily exceeded by the formatting instructions that add to the text itself. For this reason, it is a good idea to use rich text box controls primarily in connection with database fields of the MEMO or TEXT type.



In addition to the global properties, rich text box controls have the properties shown in the table below.

Specific Properties of Rich Text Box Controls	
Scrollbars	Determines if scroll bars will be displayed and if so, which type. In principle, navigation within a rich text box control is also possible without scroll bars by using the keyboard or the mouse. However, scroll bars significantly facilitate this process.

ShowAlignment	Determines whether the controls for text alignment will be shown or not.
ShowCopyPasteCut	Determines whether the controls for copying, pasting and cutting text will be shown or not.
ShowFontSelection	Determines whether the controls for basic font formatting will be shown or not.
ShowTextFormats	Determines whether the controls for text formatting will be shown or not.
ShowUndoRedo	Determines whether the Undo and Redo controls will be shown or not.

If a rich text box control is used to enter formatted text, it is also necessary to define a Docusnap rich text box in the corresponding position of the associated reports. If you do not modify the affected reports accordingly and apply a normal Docusnap text box in the reports instead, the full content of the rich text box control, including the formatting instructions in plain text, will be shown in this field. However, the reverse does not present any problems. A rich text box control can easily show the content of normal text boxes. Please note that, in this case, when editing plain text using a rich text box control, the content will automatically be assigned formatting instructions. This might lead to an erroneous presentation of the modified text when displayed in normal text boxes.

5.3.9 TreeView

Using the *TreeView* element, you can assign elements from another table to the current element using a tree structure. It is easy to assign an element from the table to be connected to the current element by enabling its checkbox. The advantage of the *TreeView* element is that you can assign multiple elements. This is helpful if the assignment via a *combo box*, which only allows a single connection, is not sufficient. To create a *TreeView* control, specify the following properties:

Specific Properties of a TreeView Control	
MappingForeignKeyField	Here, you can specify the foreign key field of the table to be used to define the relation to the table to be linked.

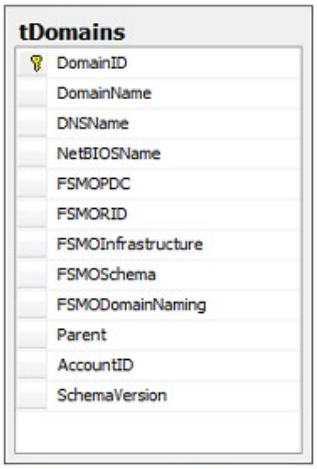
MappingReferenceField	Here, you can specify the field in the related table where the selected data from the reference table will be saved.
MappingTable	Here, you can specify the name of the table that relates both fields to be linked.
ReferencePrimaryField	Here, you can specify the primary key of the reference table.
ReferenceRecursionField	Here, you can specify an existing recursive field if the reference table is recursive.
ReferenceSQL	Here, you can specify the SQL statement used to query the required data from the reference table.
ReferenceTextField	Here, you can specify the text field to be used as a node name in the tree.

Example of use:

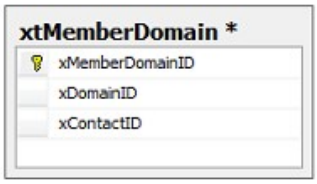
Assume you want to use the *TreeView* control to assign each contact the domain to which it belongs.

First, create a table that relates the contacts and the domains tables to each other. For this purpose, follow the instructions given in the Extending the Database Structure section.

The *xtMemberDomain* table, which includes the *xDomainID*, *xContactID* and *xMemberDomainID* fields, has the following structure:



Reference Table



Mapping Table

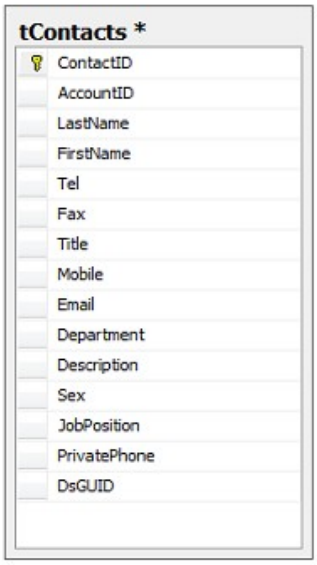
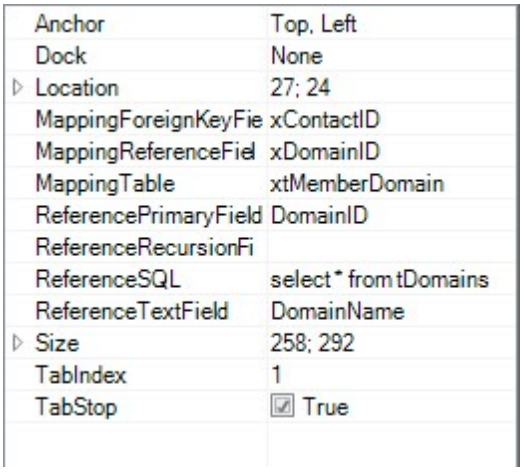


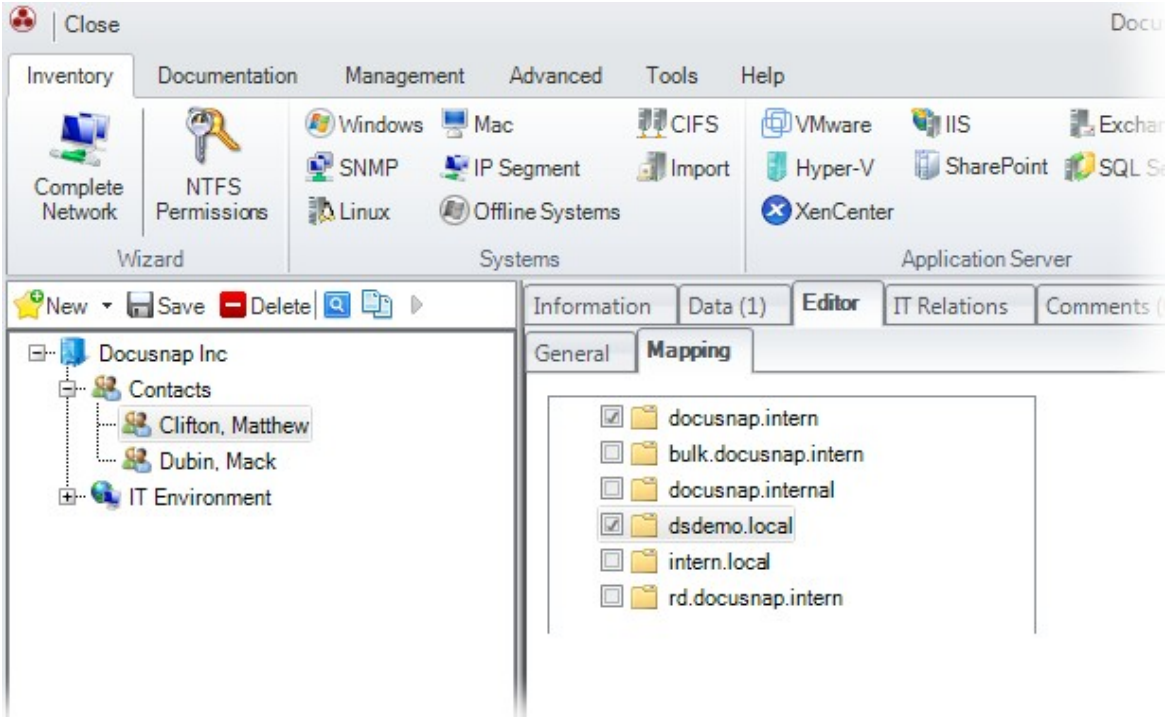
Table in Data Entry Screen

Now, all data can be specified on the Properties tab of the TreeView control.



For the ReferenceSQL property, enter a simple SELECT statement on the *tDomains* table. To obtain all domains of the current company, use the {AccountID} variable. The variable will be filled with the company AccountID for which you create this data entry screen.

After you have entered all required data, you can save the element for later use.



5.3.10 Attachments

Attachments can be added to each meta object which has a data entry screen.

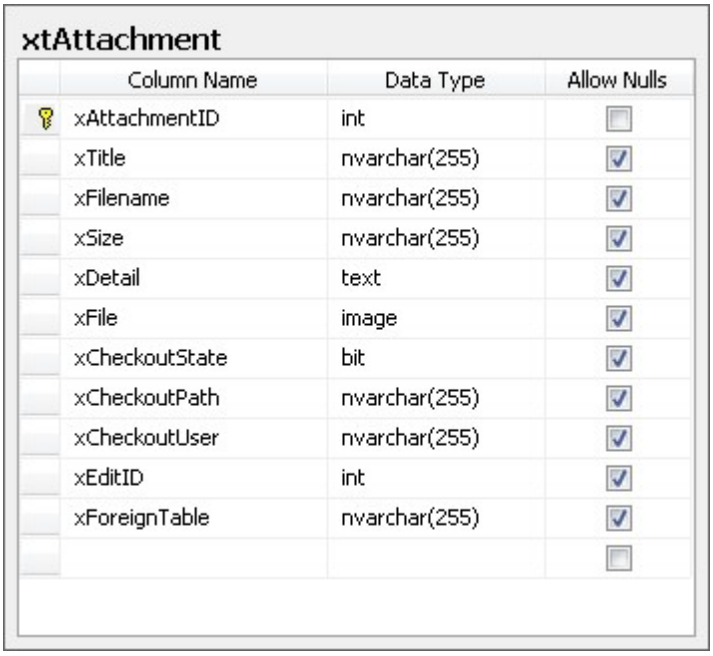
It is recommended to add a separate tab where the *Attachment* control can be inserted.

For the *Attachment* control, you must create a user-defined table in which the attachments will be saved.

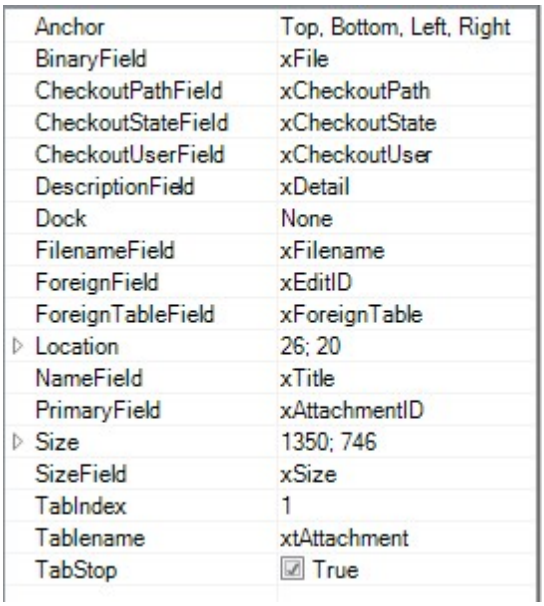
Specific Properties of Attachments	
BinaryField	Indicates the field in the database table where the file attachment will be saved.
CheckoutPathField	Indicates the field in the database table where the path for checking out the file attachment will be saved.
CheckoutStateField	Indicates the field in the database table where the checkout status of the file attachment will be saved.
CheckoutUserField	Indicates the field in the database table where the user who checked out a file attachment will be saved.
DescriptionField	Indicates the field in the database table where the description of the file attachment will be saved.

FilenameField	Indicates the field in the database table where the file name of the file attachment will be saved.
ForeignField	Indicates the field in the database table where the foreign key of the file attachment will be saved.
ForeignTableField	Indicates the field in the database table where the name of the table to which the file attachment belongs will be saved.
NameField	Indicates the field in the database table where the name of the file attachment will be saved.
PrimaryField	Indicates the field in the database table where the primary key of the file attachment will be saved.
SizeField	Indicates the field in the database table where the file size of the file attachment will be saved.
Tablename	Indicates the database table where the file attachments will be saved.

The user-defined table must contain fields that match the fields of this database table.



These fields can then be assigned to the properties of the *Attachment* control.



Anchor	Top, Bottom, Left, Right
BinaryField	xFile
CheckoutPathField	xCheckoutPath
CheckoutStateField	xCheckoutState
CheckoutUserField	xCheckoutUser
DescriptionField	xDetail
Dock	None
FilenameField	xFilename
ForeignField	xEditID
ForeignTableField	xForeignTable
Location	26; 20
NameField	xTitle
PrimaryField	xAttachmentID
Size	1350; 746
SizeField	xSize
TabIndex	1
Tablename	xtAttachment
TabStop	True

Adding Attachments

There are two ways to add attachments.

- Click the *New* button on the Attachments tab to enable the data entry screen where you can define the attachment. When you click the *Choose File* button, a dialog opens where you can select the desired file. After you have selected the desired file, the *Title* text box will automatically be populated with the file name. Optionally, you can enter a description of this attachment. To save the attachment data (*file, title, description*) in the database, click the *Save* button.
- You can also add files to the attachments by *drag & drop*. Simply drag the desired file into the table on the *Attachments* tab and drop it there. If you use *drag & drop* to add a directory or multiple files to the table, the corresponding files, or all the files in the directory, will be added simultaneously. The attachments added by *drag & drop* will immediately be stored in the database. For each added file, its file name will be used as title. You can change it and then apply your change by clicking the *Save* button.

Opening Attachments

To open an attachment in an application, first select the file. Then click the *Execute* button to open the file in the default application set for this file type in the system.

Editing Attachments

You can edit attachments whenever you want to do so. To edit the file, you must select it. Then, click the *Check Out* button. The file will be stored in the *check out path*. *Checking out* a file prevents the file from being edited by multiple users at the same time, which would introduce inconsistencies. To make the checked-out file available to other users again when you are done with editing, release it by

clicking the *Check In* button. *Checking in* the modified file saves it back to the database.

5.4 Number Servers

Functionality

The number server actually represents a special type of text box that is used to generate numbers automatically. This can, for example, be helpful to create sequential numbers for documents automatically and with a minimum of administrative effort. In addition to sequential numbers, you can specify prefixes and suffixes, as well as formatting, the starting value and an increment for the numbers.

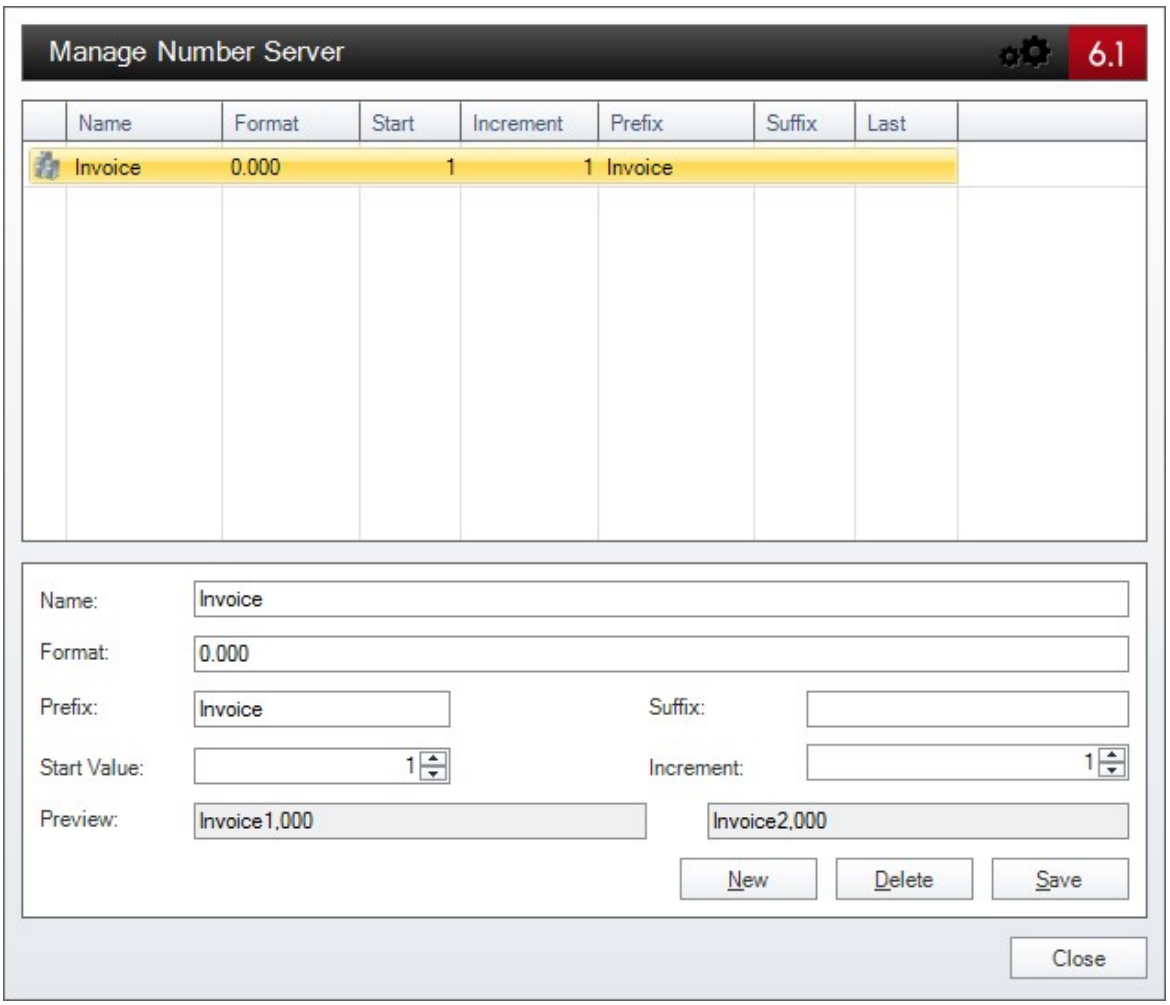
Managing the Number Server

The settings for the number server can be configured using the *Number Server* option on the *Advanced* ribbon. From the dialog that displays, it is possible to create various definitions for number servers and test them. Similar to most other management dialogs, you can create new definitions or edit or delete existing definitions. The name may be chosen freely and only serves identification purposes in Docusnap. The selected name does not affect the functionality of number servers in any way. In the Format field, enter a formatting string which determines how the sequential numbers will be presented. The formatting options behave like their equivalents provided by Microsoft for the .ToString() method in .NET, where the respective formatting string needs to be entered in quotes (""). The most important properties are explained briefly in the table below.

Character(s)	Meaning
0	Serves as a placeholder for a number or a digit of a number, where non-significant zeros will be replaced by the zero (0) character.
#	Serves as a placeholder for a number or a digit of a number, where non-significant zeros will not be replaced.
.	Inserts a decimal separator at the corresponding position.
,	Inserts a thousands separator at the corresponding position. The symbol actually used depends on the locale setting. Aside from this, the symbol divides the value by 1000. Thus, the "#,," formatting string would cause the number 1234567890 to be displayed as 1234.

%	Formats the number as a percentage and additionally multiplies it by 100. The symbol that will be used depends on the locale settings.
fp;fn;f0	Allows to specify different formats for positive and negative numbers, as well as for zero.

To specify formatting strings, you can use any desired combination of the placeholders and formatting symbols shown in the table above.

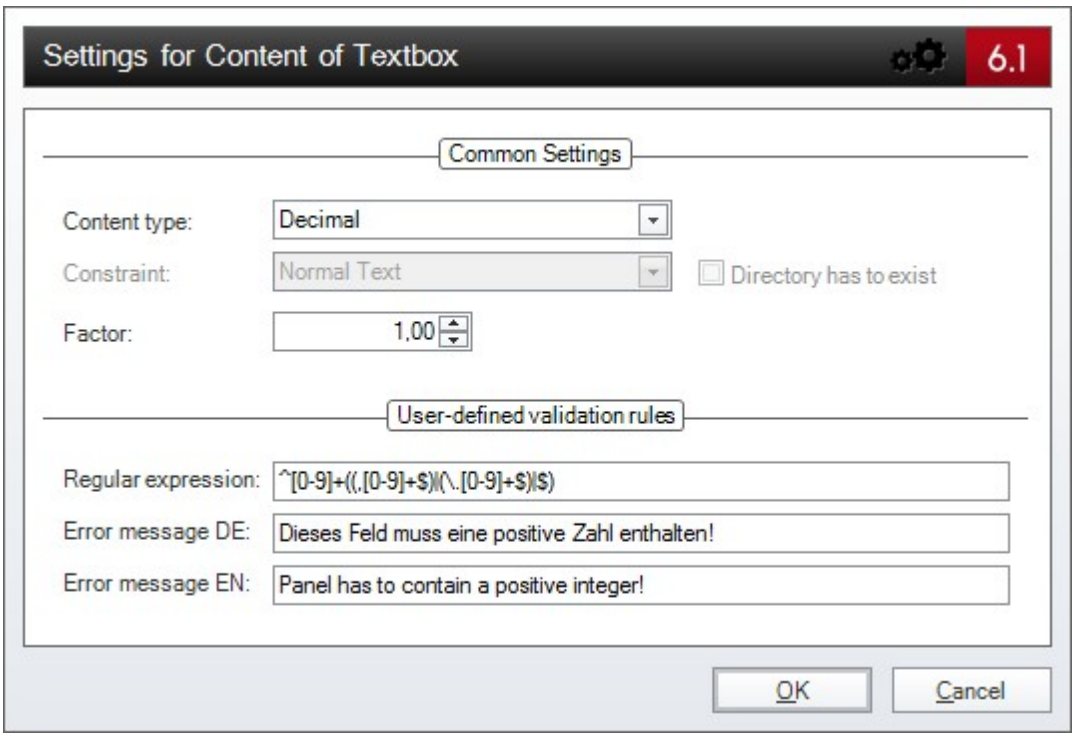


To specify a prefix that will precede the number, enter any desired string in the *Prefix* field. The *Suffix* field behaves similarly, except that this string will be shown after the formatted number. Enter the first valid value for this number definition in the *Start Value* field and specify the increment by which each subsequent number will be increased in the *Increment* field. The two text boxes at the bottom of the dialog display a preview of the first and second valid values generated. This makes it easy to see if you specified all settings correctly. When you edit the settings in

the *Manage Number Server* dialog, the preview will immediately be adjusted without requiring any additional actions.

Using the Number Server

After you have entered the corresponding definitions for the number server, they can be used in combination with the text boxes in the Designer for data entry screens. On the Properties tab of text boxes, you can set a so-called "TextConstraint" property to validate your entries. When you click the ellipsis for this property, the *Settings for Content of Textbox* dialog opens where you can select the *Number Server* content type. In the dropdown list to the right of it, you can select the previously created number server definition. After the selection has been saved, the text box will be read-only on the associated data entry screen. When a user creates a new record by means of this data entry screen, the text box will remain empty until the user clicks the *Save* button below the ribbon. If there is no value in the corresponding text box when the user saves the data, the number server determines the next value in the sequence and saves it with the record. This value will then be preset when the user accesses the record for the next time. It can no longer be modified.



Please note that settings related to the number server will only be applied when you **reload the entire data entry screen**. Since data entry screens are cached for performance reasons, it is necessary to load another data entry screen and then switch back to the modified data entry screen to do so. For this purpose, it is not sufficient to switch to

one of the extensions or to display a list view. The entire data entry screen must be reloaded. Example: After having customized the company data entry screen, you could load the contacts data entry screen and then switch back to the company data entry screen.

5.5 Example

Opening the Designer

As a simple example of a full user-defined data entry screen, this section will explain how to create a suitable user interface for the SLA object already explained in the preceding sections. For this purpose, first select the SLA caption in the Data Explorer and then create a new SLA data object by clicking the *New* button above the tree view. However, since a user interface does not yet exist for this object, you need to define it by clicking the *Design Mode* button on the *Tools* ribbon. Once the Designer is open, the workspace in the right pane will be marked with an orange border.

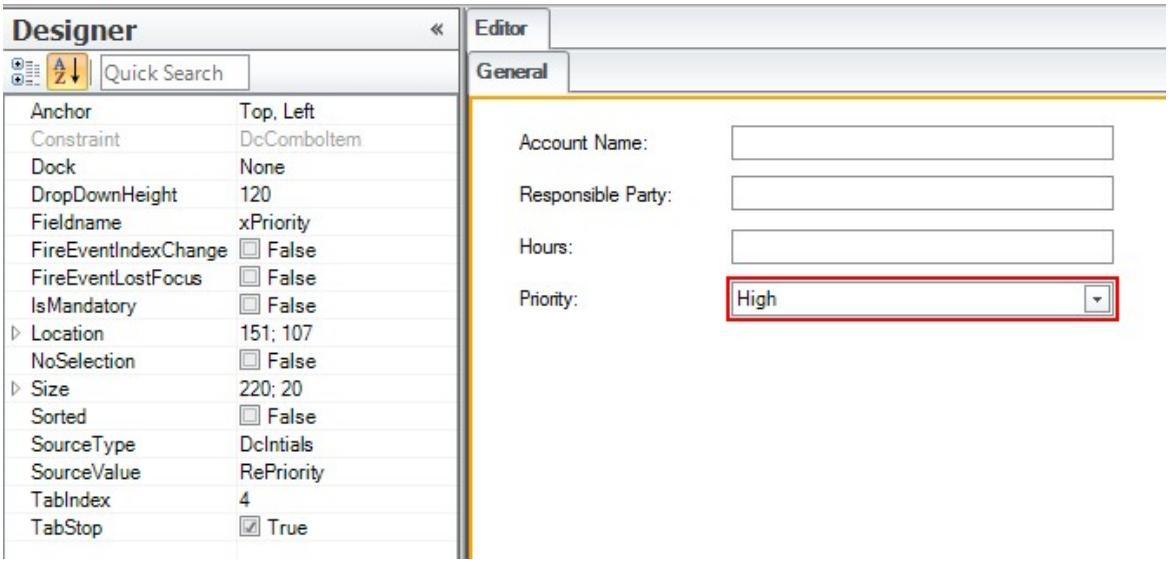
Designing the User Interface

In order to be able to enter all data that is required to create a new SLA object, a total of eight controls are necessary which will be briefly introduced in the table below, including their properties. The *Control* column indicates the type; the *Field Name* column the value of the *Fieldname* property; and the *Notes* column shows other required settings.

Control	Field Name	Characteristics
Label		Entry: Name
Label		Entry: Responsible Party
Label		Entry: Hours
Label		Entry: Priority
Text Box	xName	TabIndex: 1; IsMandatory: True
Text Box	xResponsible	TabIndex: 2
Text Box	xHours	TabIndex: 3, TextConstraint: Float

Combo Box	xPriority	TabIndex: 4; SourceType: DcInitials; SourceValue: REPRIORITY

Once you have added all eight controls, the user interface should appear as shown in the figure below and have all required features. All four data entry controls are connected automatically to the corresponding columns in the Docusnap database as you specified the *Fieldname* property. The combo box will be filled with a predefined list of different selectable priority settings, because you assigned the REPRIORITY reference value. For more information about the other available reference values, refer to the appendix of this manual.



If you added and configured all controls properly on the Properties tab, you can save the new data entry screen by clicking the *Save Configuration* button in the lower part of the toolbox. Depending upon the active configuration, the data entry screen will be stored in the folder for local or team settings. If you want that other users have access to the new data entry screen too, you need to distribute the corresponding file to these users. For more information on this topic, refer to the Distributing Customizations section below.

Testing the Newly Created Data Entry Screen

To test if the newly created data entry screen works properly, close the Designer by clicking the *Design Mode* button on the *Tools* ribbon again. If unsaved changes exist, you will be prompted to save them. Now, it is possible to create a new SLA object by selecting the SLA caption and clicking the *New* button above the Data Explorer tree view. The newly created data entry screen will appear in the main window of Docusnap and users can enter and edit data as required. By clicking the *Save* button above the Data Explorer tree view, the data is saved directly in the current or new record of the user-defined Docusnap table.

Please note that errors that occur during this test are not necessarily caused by errors in the data entry screen. Due to the fact that this SLA example consists of three components, errors may arise from any one of the components and might only now become visible. For example, if data input controls are shown in read-only mode, it is very likely that Docusnap was not able to establish the connection to the database. In most cases, this error is due to incorrect or missing settings for the *Fieldname* property of the respective controls.

Part VI

6 Distribution of Modifications

Underlying Principle

In Docusnap, it is possible to export customizations made to the database structures, the meta objects and the data entry screens and to apply them to other databases and Docusnap installations without much effort by importing them in the other environment.

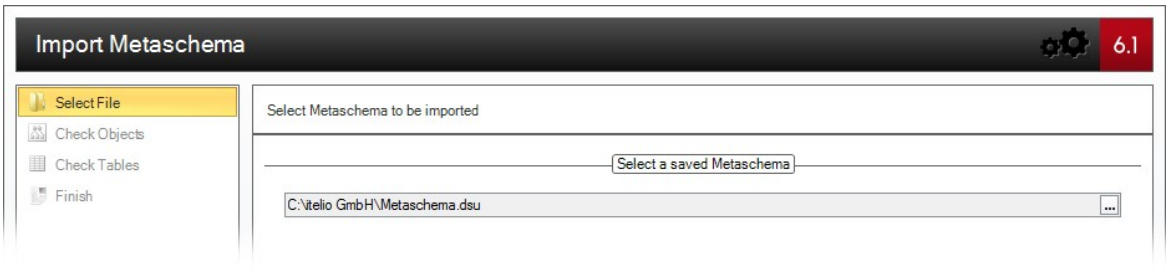
Exporting the Database Structure and the Meta Objects

In order to export a meta schema, Docusnap must be connected to the database that contains the corresponding schema. The current schema can be exported to any desired location by clicking the *Export Schema* button on the *Tools* ribbon of the main Docusnap window. At the selected location, Docusnap creates a new file with the .dsu file extension which contains all customizations made to the current Docusnap database. Please note that a partial export of the customizations is not possible.

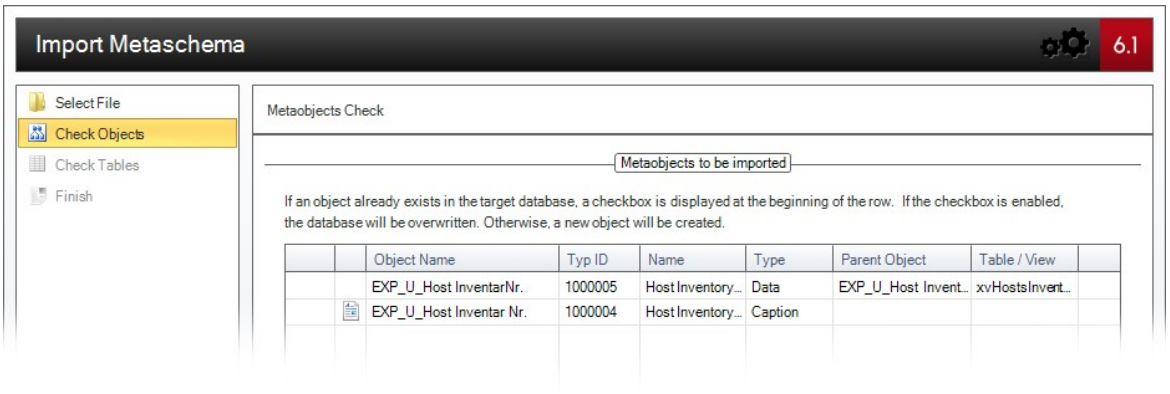
Importing the Database Structure and the Meta Objects

Previously exported customizations of the meta schema can be imported into another database using a wizard. To do so, Docusnap needs to be connected to the target database at the time of the import process. By clicking the *Import Schema* button on the *Tools* ribbon of the Docusnap main window, you can open the associated wizard that helps you specify further settings.

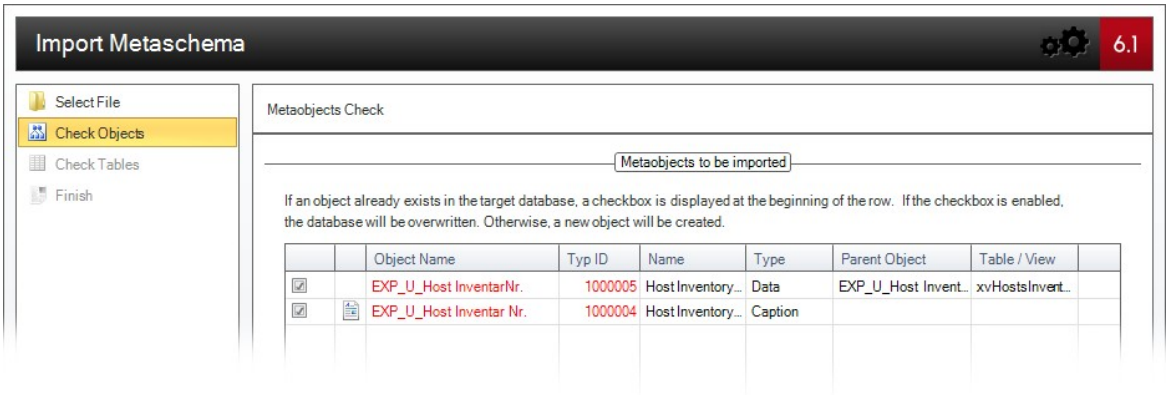
The first step in the wizard is to select the previously exported file which contains the modified meta schema. When you click the ... button, a file selection dialog opens where you can open the desired .dsu file.



In the second step, the wizard will display all customized and newly added meta objects that exist in the selected file and that will be imported into the current database. When importing the file, Docusnap considers all objects from the schema file. This means that all meta objects contained in the file will be imported. Due to the potential dependencies between the objects, a selection is not possible at this time. After the import, you can use the *Manage Metaobjects* dialog to delete any objects that are not required.



If a checkbox is displayed next to an object to be imported, an object with the same type ID already exists in the target database. The red font indicates that the corresponding object name or type ID is already present among the meta objects of the target database. If you enable the checkbox, the corresponding object in the target database will be overwritten by the information from the selected file. If you leave the checkbox unchecked, a new object with the same name, but a different type ID, will be created in the target database. In this case, the original object in the target database remains unchanged. If the type ID already exists but the object name is different, and you select the option for overwriting existing data, Docusnap will also overwrite the existing object name with the one from the import file.

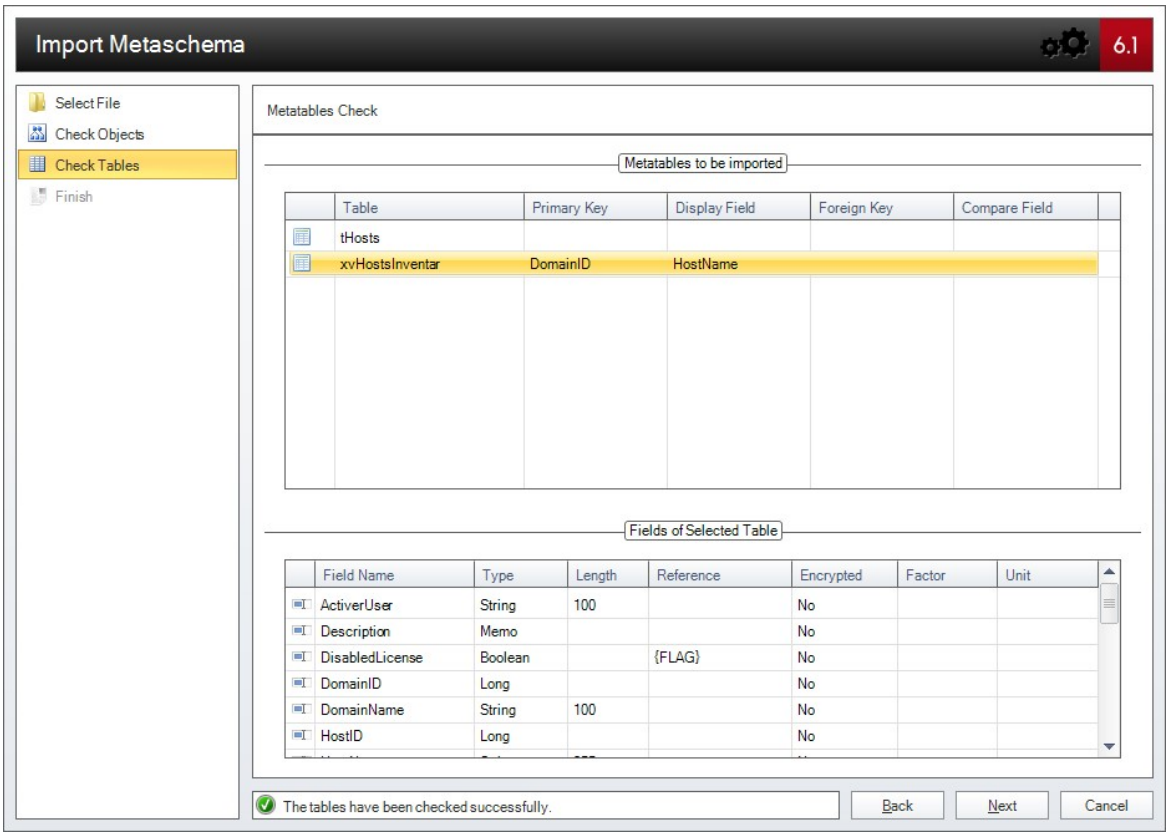


The third step of the wizard lists all tables that are present in the selected file and that will be imported into the target database. The following figure shows a table called *xtSLA* which does not exist in the Docusnap system schema. The *tHosts* table, in contrast, belongs to the system schema. A field was added to it where users can enter additional information.

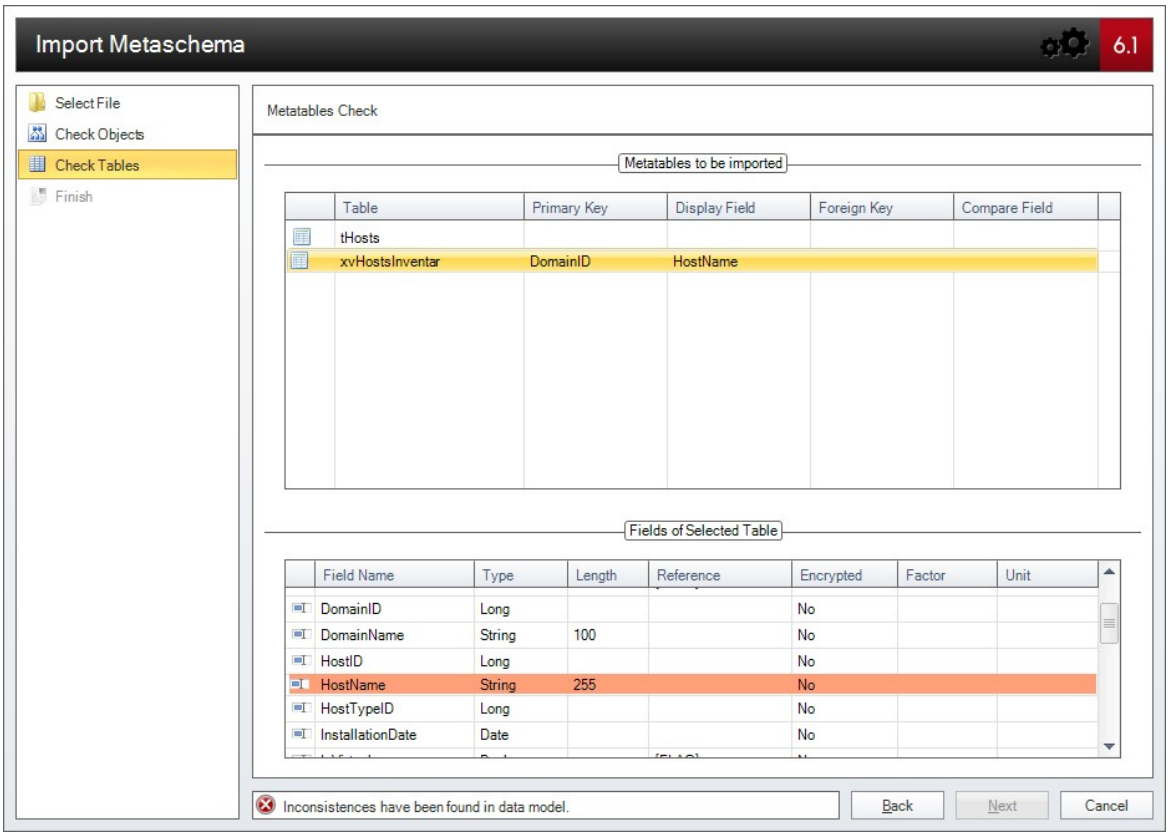
The *Metatables to be imported* list displays all tables that are either user-defined or have been customized by adding user-defined fields or by editing existing fields. When you select a table in the upper list, all added or modified columns of the selected table will be displayed in the *Fields of Selected Table* list below.

If a table does not yet exist in the target database, the table will be created with all its fields. If the table already exists, this list shows only the columns that are still missing. Docusnap will never delete fields that, while existing in the database, are

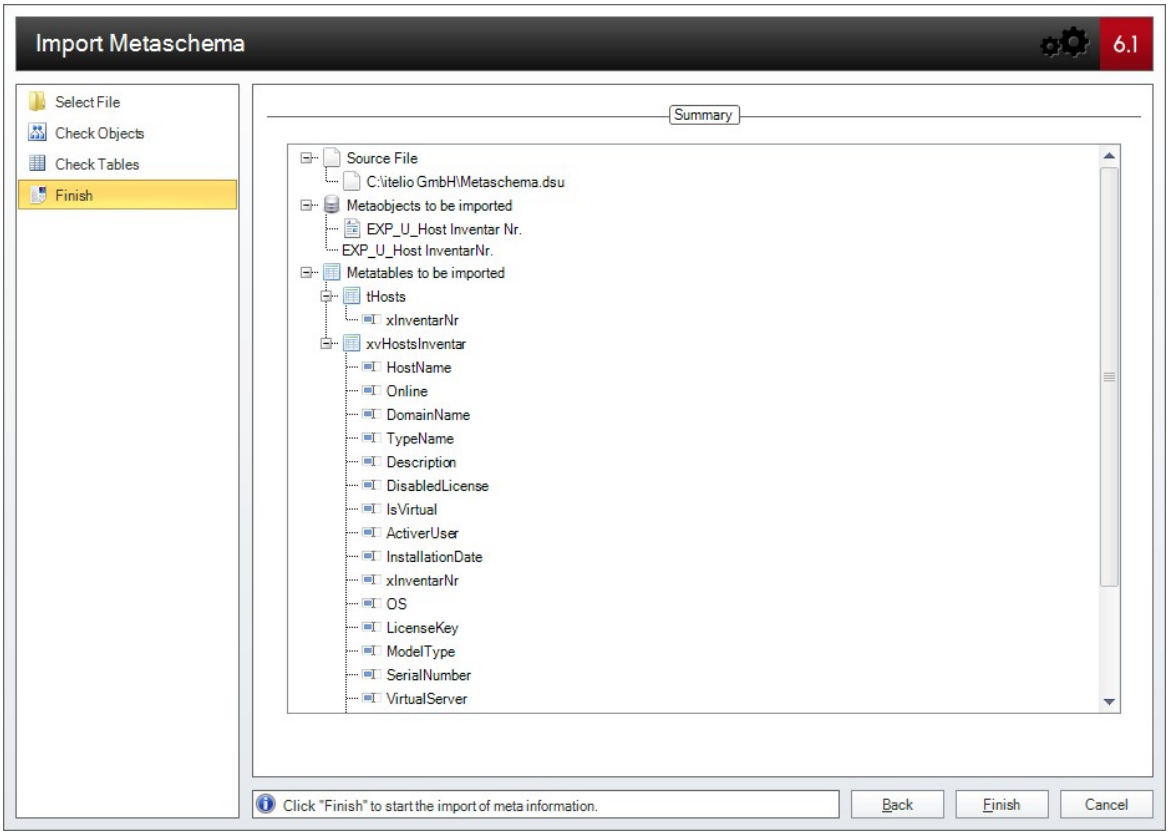
not associated with that table in the schema file. All tables and fields will be imported during the import process. It is not possible to exclude individual tables or fields from being imported.



If a field already exists in the target database, but has a different data type than that of the definition in the import file, the field will be highlighted in red in the corresponding list. When you click the table in the upper list, Docusnap will display the columns in the lower list, highlighting the fields that still have problems in red. To perform the import process, change the data type in either the target database or the source database. However, the data type can only be changed by deleting the affected column and re-creating it with the other data type. Please note, however, that all data in this field will be deleted from the database. If you change the field in the source database, you need to re-export the schema file to make sure that the updated data will be imported. If you change the field in the target database, however, it is sufficient to only delete the field, since it will be re-created by importing the source file, this time with the appropriate data type from the source database.



The last step of the wizard shows a summary of all objects and tables to be imported. By clicking the *Back* button, you can change the selection, if required. To import the objects and tables into the target database, click the the *Finish* button.



Distributing Data Entry Screens

Customized or newly created data entry screens are not automatically distributed by means of a wizard, since they are independent from the database in use. Instead, they will be loaded from the respective local or team settings.. In order to make these changes available to multiple users, these users need access to the corresponding .dsu files. If team settings are used, it is usually sufficient to copy the modified .dsu files to the *DataEdit* subdirectory of the team settings directory, unless this has been done automatically when saving the data. If local settings are used, the corresponding files will always have to be distributed manually.

Even if team settings are used, changes made to the data entry screens might not immediately be loaded into all Docusnap installations. In particular, if Docusnap is used on laptop computers by field representatives who are not connected to the corresponding team settings directory, the local settings will be used alternatively. In this case, manual distribution of the changes is indispensable in order to enable all users to use the modified data entry screens.

Part VII

7 Appendix

This appendix contains information that adds to the manual.

7.1 Reference Values

Name	Filter Value	Description
Access control type	AccessControlType	Grant and deny ACL and ADS permissions
ACE types	ACETypes	Allow and Deny ACE types
Enabled / Disabled	ActivatedDeactivated	Enabled and Disabled values, if required
Address list types	AddressListType	Address types in tExchAddressLists
ADS properties data type	ADSProperty	Data type for ADS properties
Maximum number allowed	AllowMaximum	Maximum number allowed
Architecture	Architecture	Architecture in tProcessor
Type of attachment	AttachmentType	Type of attachment
Automatic update options	AUOptions	Automatic update options in tDocuWindows
Yes/No	Bool	Yes and No values
Boot method	BootMethod	Boot method for DNS servers in tDNSServerDocu
Page break	BreakType	Page break type in IT concepts
Caption type	CaptionType	Caption type in IT concepts

Name	Filter Value	Description
Chassis Types	ChassisTypes	Various Chassis Types
Combo box data source	ComboBoxDataSource	List of options for combo boxes in data entry screens
Configuration status	ConfigStatus	Configuration status in tESXVirtualMachine
Connection status	ConnectionState	Connection status in tESXVirtualMachine
Country	Country	List of countries
CPU limit	CPULimit	CPU limit, if unlimited; otherwise the value is taken from the database
CPU status	CPUStatus	CPU status
Database type	DatabaseType	Database type
Data types	DataTypes	Data type in tSQLColumns
Debug information	DebugInfo	Debug information
Debug level	DebugLevel	Debug level
Debug target	DebugTarget	Debug target
AU check interval activation	DetectionFrequencyEnabled	Automatic update check interval activation in tDocuWindows
DHCP active lease types	DHCPActiveLeasesTypes	DHCP active lease types
Scope options	DHCPScopeOptions	Scope options for DHCP

Name	Filter Value	Description
Diagram types	DiagramType	Diagram types in meta objects
Data source	DocDataSource	Data source in IT concepts
Catalog element type	DocuCatalogElements	Catalog element type in IT concepts
Data element source	DocuDataelementSource	Data element source in IT concepts
Directory element	DocuToCElements	Directory element in IT concepts
Drive types	DriveTypeCollection	Drive types in tDrives
Enabled / Disabled	EnabledDisabled	Values for Enabled and Disabled
Asset type	ESXAssetType	Asset type in tESXLicenseAsset
ESX Color State	ESXColorState	Status information by color
ESXComputeResource types	ESXComputeResource	Types in tESXComputeResource
ESX Power Management	ESXDpmBehavior	ESX Power Management
ESX DRS Managment	ESXDrsBehavior	ESX DRS Managment
Feature status	ESXFeatureState	Feature status in tESXFeature
HT Sharing	ESXHtSharing	HTSharing in tESXVmRessource
HT sharing	ESXHtSharing	HTSharing in tESXVmRessource
HA host isolation response	ESXIsolationResponse	High Availability host isolation response

Name	Filter Value	Description
ESXLevel	ESXLevel	Share level, CPU level, RAM level in tESXVmHwDisk & tESXVmRessource
CPU/ MMU virtualization	ESXMMU	CPU/ MMU virtualization in tESXVmOptions
Accept / Reject	ESXPortGroupStatus	Accept and Reject values
VM Restart Priority	ESXRestartPriority	Virtual Machine Restart Priority
CPU and RAM usage	ESXSharesLevel	CPU usage as a percentage, RAM usage as a percentage in tESXResourcePool
ESX power management	ESXStandby	Power management in tESXVmOptions
ESX swap file location	ESXSWAP	Location of swap file in tESXVmOptions
ESX swap file location	ESXSwapPlacementType	Location of swap file in tESXComputeResource
ESX tool upgrade	ESXToolUpgrade	Tool upgrade method in tESXVmOptions
ESX virtual disk mode	ESXVirtualDiskMode	Virtual disk mode in tESXVmHwDisk
VM Monitoring	ESXVmMonitoringState	Virtual Machine Monitoring
Domains accepted in Exchange	ExchangeAcceptedDomains	Domains accepted in Exchange
Exchange SRV certificate	ExchangeCertificateAuthorityType	Exchange SRV certificate

Name	Filter Value	Description
Exchange SRV certificate service	ExchangeCertificateServices	Exchange SRV certificate service
Exchange SRV certificate status	ExchangeCertificateStatus	Exchange SRV certificate status
Exchange database types	ExchangeDatabaseType	Database types in Exchange
Exchange distribution group types	ExchangeDistributionGroupType	Exchange distribution group types
Exchange distribution group recipient types	ExchangeDistributionGroupRecipientType	Exchange distribution group recipient types
Exchange e-mail policy priority	ExchangeEmailPolicyPriority	Only for the "Lowest" priority, for all other priorities, a number is used
Permissions for Exchange mailboxes	ExchangeMailboxAccessRights	Permissions for Exchange mailboxes
Exchange mailbox folder types	ExchangeMailboxFolderType	Exchange mailbox folder types
Exchange mailbox types	ExchangeMailboxTypes	Exchange mailbox types
Exchange managed folder types	ExchangeManagedFolderTypes	Exchange managed folder types
Exchange object type	ExchangeObjectType	Exchange object type

Name	Filter Value	Description
Exchange protocol logging level	ExchangeProtocolLoggingLevel	Exchange protocol logging level
Permissions for Exchange public folders	ExchangePublicFolderAccessRights	Permissions for Exchange public folders
Exchange server roles	ExchangeServerRoles	Exchange server roles
Exchange version	ExchangeVersion	Exchange version
Processor family	Family	Processor family in tProcessor
Depreciation method	FiDepMethod	Depreciation method in tExFinance
Meta table data type	FieldType	Data type of the columns in meta tables
File system	FileSystem	File system in tDrives
Financial costs	FinanceCost	Financial costs in tExFinance
Payment interval	FinanceInterval	Interval in which payments will be made in tExFinance
Financing type	FinanceType	Financing types in tExFinance
Yes/No	Flag	Values for Yes and No
Form factor	FormFactor	Form factor in tRAM
Exchange Permissions	FriendlyExchangeRights	Exchange Permissions
NTFS permissions	FriendlyNTFSPermission	NTFS permissions

Name	Filter Value	Description
SharePoint permissions	FriendlySharePointPermission	SharePoint permissions
IT Assets Field Type	GOFieldType	Field Type for IT Assets
ESX guest status	GuestState	Guest status in tESXVirtualMachine
Hyper-V automatic shutdown action	HVAutomaticShutdownAction	Hyper-V automatic shutdown action
Hyper-V automatic startup action	HVAutomaticStartupAction	Hyper-V automatic startup action
Hyper-V boot device order	HVBootOrderDevice	Hyper-V boot device order
Hyper-V status	HVEnabledSate	Hyper-V status
Hyper-V heartbeat	HVHealthState	Hyper-V heartbeat
Hyper-V integration service status	HVIntegrationEnabledState	Hyper-V integration service status
Hyper-V virtual computer status	HVOperationalStatus	Hyper-V virtual computer status
Hyper-V integration service	HVServiceType	Hyper-V integration service
IIS notification type	IISBITSServerNotificationtype	IIS notification type

Name	Filter Value	Description
IIS directory security	IISDirectorySecurity	IIS directory security
IIS server status	IIServerState	IIS server status
IIS web service extension status	IISExtensionState	IIS web service extension status
IIS filter status	IISFilterState	IIS filter status
IIS log schedule	IISLogFilePeriod	IIS log schedule
IIS directory content	IISNntpUseAccount	IIS directory content
IIS smart host type	IISSmartHostType	IIS smart host type
IIS FTP user isolation mode	IISUserIsolationMode	IIS FTP user isolation mode
Image alignment	ImageAlignment	Image alignment in IT concepts
Image source	ImageSource	Image source in IT concepts
Inheritance flag	InheritanceFlag	Inheritance flag in tADSRights
Inheritance type	InheritanceType	Inheritance type in tADSRights
Install State	InstallState	Install State of Optional Features
Interleave position	InterleavePosition	Interleave position in tRAM
User-defined	IsUserDefCollection	Has the computer been defined by the scan process or manually
Exchange journal rules	JournalRuleTypes	Exchange journal rules

Name	Filter Value	Description
Keyword type	KeywordType	Keywords for licenses in tSoftwareLicenses
Language	Language	Language
License assignment	LicenseMapping	License assignment in tSoftwareProductTypes
Lines	LineStyleCollection	Lines
Linux file systems	LinuxFileSystem	Linux file systems
Linux group status	LinuxLocalGroupsStatus	Linux group status in tLinuxLocalGroups
Linux systems	LinuxSystem	Linux systems
List alignment	ListAlignment	Alignment of meta object lists
Local groups / local user status	LocalAccountsStatus	Local groups / local user status
Overwrite log	LogOverwrite	Overwrite log in tDocuWindows
Mac file systems	MacFileSystem	Mac file systems
Mac systems	MacSystem	Mac systems
Unit of measure	MeasuringUnit	Unit of measure in IT concepts
Memory type	MemoryType	Memory type in tRAM, tVideoController
Document type	MetaDocumentType	Document type in IT concepts
Modules	Module	Modules
NetBIOS option	NetbiosOptions	NetBIOS option

Name	Filter Value	Description
Connection status	NetConnectionStatus	Status in tNetworkAdapter
Linux protocol	NetServicesProtocol	Net protocol in tLinuxNetServices
Nic Teaming	NicTeamingPolicyType	ESX Nic Teaming
Automatic update	NoAutoUpdate	Automatic update in tDocuWindows
NetBIOS node type	NodeType	NetBIOS node type
Allowed number of users	NumberOfUsersShares	Allowed number of users in tShares
Meta object categories	ObjectCategory	Categories of meta objects
On/Off	On/Off	Values for On and Off
Connected / Not connected	OnlineCollectionYesNo	Values for Connected / Not connected
Online status	OnlineStatus	Online status of the computer in tHosts
Operating system	OSType	Operating system
Overall status	OverallStatus	Overall status in tESXComputeResource, tESXDataCenter, tESXDataCenterFolder, tESXDataCenterHostFolder, tESXHost & tESXVirtualMachine
Page number format	PageNumberFormat	Page number format in IT concepts

Name	Filter Value	Description
Password logging	PasswordLogging	Button used for password logging
Password type	PasswordType	Password type in tExPassword
Password validity	PasswordValid	Password validity in tExPassword
Permission	Permission	Permission in tSharePermission
Permission type	PermissionType	Permission type in tSharePermission
Position	Position	Position
Power options	PowerPolicyNever	Power options default if no option is selected.
Virtual machine status	PowerState	Status in tESXVirtualMachine
Product suite	ProductSuite	Product suite in tOSProductSuite
Propagation flags	PropagationFlags	Propagation flags in tADSRights
Supported SCSI protocols	ProtocolSupported	All protocols supported in tSCSI
RAM limit	RAMLimit	RAM limit in tESXVmRessource
Priority	RelPriority	Priority in tRelLink
ADS connection type	ReplicationType	Type in tADSConnection
Report formatting	ReportAlignmentHorizontal	Report formatting
Report formatting	ReportAlignmentVertical	Report formatting

Name	Filter Value	Description
Report formatting	ReportComponentStyleCollection	Report formatting
Report formatting	ReportFillStyleCollection	Report formatting
Display cover page, header and footer	ReportPartState	Display cover page, header and footer
Display comment in the report	ReportPosition	Comment will be shown in the report
Priority	RePriority	Priority in tExReminders
CPU and RAM reservation types	ReservationType	Reservation types for CPU and RAM in tESXResourcePool
Task scheduling type	ScheduleActionType	Task scheduling type
Author of scheduled task	ScheduledTaskAuthor	Author of scheduled task
Status of scheduled task	ScheduledTasksStatus	Status of scheduled task
Server Features	ServerFeatures	Server Features
Service start type	ServiceStartType	Service start type in tServices
Service status	ServiceStatus	Status in tServices
Gender	Sex	Gender in tContacts
Share type	ShareType	Type in tShares

Name	Filter Value	Description
SID type	SIDType	SID type in tLocalAccounts, tLocalGroups & tLocalGroupMembers
Snapshot status	SnapshotStatus	Snapshot status in tESXVirtualMachineSnapShots
SNMP mapping	SNMPDeviceMappingType	SNMP mapping
SNMP interface type	SNMPInterface	Type in tSNMPInterface
SNMP interface connection	SNMPInterfaceConnection	Connection in tSNMPInterface
SNMP NetToMedia type	SNMPNetToMediaType	Type in tSNMPNetToMedia
SNMP type	SNMPType	SNMP type in tSNMPTypes
SNMP type	SNMPVaribleType	SNMP type in connection with an MIB
SharePoint authentication	SPAuthentication	SharePoint authentication
SharePoint authentication provider	SPAuthenticationProvider	SharePoint authentication provider
SharePoint permissions	SPBasePermissions	SharePoint permissions
SharePoint list type	SPBaseType	SharePoint list type
SharePoint browser file handling	SPBrowserFileHandling	SharePoint browser file handling

Name	Filter Value	Description
SharePoint administrator permissions	SPCentralAdministrationRights	SharePoint administrator permissions
SharePoint content source type	SPContentSourceType	SharePoint content source type
SharePoint content source priority	SPCrawlPriority	SharePoint content source priority
SharePoint crawl rule authentication type	SPCrawlRuleAuthenticationType	SharePoint crawl rule authentication type
SharePoint days	SPDaysOfMonth	SharePoint days for schedule output
SharePoint days of the week	SPDaysOfWeek	SharePoint days of the week for schedule output
SharePoint deployment server type	SPDeploymentServerType	SharePoint deployment server type
Show SharePoint draft items list	SPDraftVisibilityType	Show SharePoint draft items list
Network adapter duplex mode	SpeedDuplex	Network adapter duplex mode
SharePoint - federated location authentication information	SPFederationAuthType	SharePoint - federated location authentication information

Name	Filter Value	Description
SharePoint template category	SPListCategoryType	SharePoint template category
SharePoint federated location	SPLocationType	SharePoint federated location
SharePoint website collection lock status	SPLockStatus	SharePoint website collection lock status
SharePoint months	SPMonthsOfYear	SharePoint months for schedule output
SharePoint status	SPObjectStatus	SharePoint status
SharePoint action	SPOfficialFileAction	SharePoint action
SharePoint code page	SPOutboundMailCodePage	SharePoint code page
Managed SharePoint paths type	SPPrefixType	Managed SharePoint paths type
SharePoint role type	SPPrincipalType	SharePoint role type
SharePoint read access	SPReadSecurity	SharePoint read access
SharePoint request frequency	SPRuleBehavior	SharePoint request frequency
SharePoint Running Job	SPRunningJobStatus	SharePoint Running Job Status

Name	Filter Value	Description
Status		
SharePoint schedule type	SPScheduleType	SharePoint schedule type
SharePoint search scope rule behavior	SPScopeRuleFilterBehavior	SharePoint search scope rule behavior
SharePoint scope rule type	SPScopeRuleType	SharePoint scope rule type
SharePoint URL rule type	SPScopeUrlType	SharePoint URL rule type
SharePoint server role	SPServerRole	SharePoint server role
SharePoint solution deployment	SPSolutionDeploymentState	SharePoint solution deployment
SharePoint solutions - result of last operation	SPSolutionOperationResult	SharePoint solutions - result of last operation
SharePoint time zone	SPTimeZone	SharePoint time zone
SharePoint URL zone	SPUrlZone	SharePoint URL zone
SharePoint view audience	SPViewAudience	SharePoint view audience
SharePoint indexing mode for ASPX pages	SPWebASPXPageIndexMode	SharePoint indexing mode for ASPX pages

Name	Filter Value	Description
SharePoint weeks of the month	SPWeeksOfMonth	SharePoint weeks of the month for schedule output
SharePoint Write access	SPWriteSecurity	SharePoint Write access
SQL Class	SQLClass	SQL Class
SQL Compatibility	SQLCompatibility	Compatibility in tSQLDatabases
SQL Procedure Type	SQLProcType	SQL Procedure Type
SQL Status	SQLState	Status of SQL Permissions
SQL Table Type	SQLTableType	SQL table type in tSQLTables
SQL User Type	SQLUserType	SQL User Type
Hardware status	Status	Hardware status in tRAM, tProcessor, tSCSI, tCDRomDrive, tSoundDevice, tDesktopMonitor, tVideoController, tTapeDrive & tNIC
Storage Adapter Type	StorageAdapterType	ESX Storage Adapter Type
Storage Adapter Device Type	StorageLunType	ESX Storage Adapter Device Type
Storage Adapter Path Status	StoragePathState	ESX Storage Adapter Path Status
Table type	TableType	Table type in tSQLTables
Task scheduling	TaskSchedulerLibrary	Task scheduling

Name	Filter Value	Description
Timeout option	TimeoutOption	LAN or WAN timeout setting
Time zones	TimeZone	Time zones
VMware tool status	ToolsStatus	VMware tools in tESXVirtualMachine
Sort order	TreeSortOrder	Sort order in meta objects
Tree type	TreeType	Tree type
True/False	True/False	True and False values
Trust relationship	TrustType	Trust relationship
Type details for RAM	TypeDetail	Type specifics in tRAM
Language	UILanguage	Language
Upgrade method	UpgradeMethod	Upgrade method in tProcessor
USB status	USBStatus	USB status in tUSB
User limitation	UserLimitation	User limitation in tShares
Memory type	VideoMemoryType	Memory type in tVideoController
Device type	VMDeviceType	Device type in tESXVmHwDevices
VMware machine status	VMSnapShotState	Status of a virtualized VMware instance
Windows file systems	WindowsFileSystem	Windows file systems
Windows systems	WindowsSystem	Windows systems

Name	Filter Value	Description
Wizards	WizardOptions	List of wizards available for dashboards
Types for XenCenter	XenAfterApplyGuidance, XenBondMode, XenDuplex, XenHostAllowedOperations, XenIpConfigurationMode, XenNetworkOperations, XenOnCrashBehaviour, XenOnNormalExit, XenStorageOperations, XenVbdOperations, XenVdiOperations, XenVifOperations, XenVmApplianceOperation, XenVmOperations, XenVmPowerState	Types specific for XenCenter